

IMPLEMENTACIÓN DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN
EN EL PROCESO DE CONTROL DE INVENTARIOS EN LA EMPRESA A.T.I.S.
(ASESORES EN TI & SOFTWARE)

LINA MARIA CUBILLOS VERGARA

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ D.C
2014

IMPLEMENTACIÓN DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN
EN EL PROCESO DE CONTROL DE INVENTARIOS EN LA EMPRESA A.T.I.S.
(ASESORES EN TI & SOFTWARE)

LINA MARIA CUBILLOS VERGARA

TRABAJO DE TESIS DE GRADO PARA OBTENER EL TÍTULO PROFESIONAL
DE INGENIERO DE SISTEMAS

Director de Proyecto

JORGE ELIECER VARGAS PUERTO

Ingeniero de Sistemas

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ D.C
2014

NOTA DE ACEPTACIÓN

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bogotá D.C

A Dios, a la vida, y a mi familia, por una nueva oportunidad...

AGRADECIMIENTOS

A Dios, por permitirme gozar de esta experiencia, a mi madre y mi padre, Gloria Vergara y Antonio Cubillos quienes son mi mejor ejemplo, mi mayor orgullo y a quien les debo este nuevo triunfo, y a todos aquellos que con su amor y apoyo me permitieron hacer de un sueño realidad... Gracias.

CARTA DE SESIÓN DE DERECHOS

Señores:

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

Bogotá D.C.

Estimados señores:

Yo Lina Maria Cubillos Vergara, identificada con Cédula de ciudadanía No. 1.069.734.706 de Fusagasugá, autora de la tesis de grado titulado,

“IMPLEMENTACIÓN DE SISTEMA DE INFORMACIÓN PARA LA OPTIMIZACIÓN EN EL PROCESO DE CONTROL DE INVENTARIOS EN LA EMPRESA A.T.I.S. (ASESORES EN TI & SOFTWARE)”,

Tesis presentada como requisito para obtener el título de ingeniero de sistemas, autorizo a la Fundación Universitaria Los Libertadores la consulta, reproducción, distribución o cualquier otra forma de uso de la obra parcial o total, con fines académicos en cualquier formato de presentación; conforme a la ley 23 de de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995, Circular No 06 de la Dirección Nacional de Derechos de Autor para las Instituciones de Educación Superior, y demás normas generales en la materia.

Firma.....

Nombre.....

C.C. No.....de.....

CONTENIDO

	Pág.
INTRODUCCIÓN	10
OBJETIVOS	11
1. ASPECTOS DE LA INVESTIGACIÓN	12
1.1 PLANTEAMIENTO DEL PROBLEMA	12
1.2 FORMULACIÓN DEL PROBLEMA	12
1.3 CONDICIONES DEL SISTEMA ACTUAL	12
1.4 DEFINICIÓN DEL SISTEMA PROPUESTO	13
1.5 JUSTIFICACIÓN.	13
1.6 DELIMITACIÓN	14
1.7 ALCANCE.	15
1.8 ESTUDIOS DE FACTIBILIDAD TÉCNICA, OPERATIVA Y ECONÓMICA	16
1.8.1 TÉCNICA.	16
1.8.2 OPERATIVA.	17
1.8.3 ECONÓMICA.	17
2. MARCO REFERENCIAL	18
2.1 MARCO HISTÓRICO	18
2.2 MARCO TEÓRICO	19
2.2.1. SCRUM	19
2.2.2. FASES DE SCRUM	20
2.3 MARCO CONCEPTUAL	22
2.3.1 SISTEMA DE INFORMACIÓN	22
2.3.2 COMPUTACIÓN EN LA NUBE	22
2.3.3 ARQUITECTURA DE SOFTWARE	22
2.3.4 APLICACIÓN WEB	22
2.3.5 WINDOWS AZURE	23
2.3.6 SERVICIO WEB	23
2.3.7 MANEJADOR DE BASES DE DATOS	23

3. DISEÑO METODOLÓGICO	24
3.1 REQUERIMIENTOS FUNCIONALES	24
3.1.1 REQUERIMIENTO DE ATENTICACIÓN Y SEGURIDAD	24
3.1.2 REQUERIMIENTO CARGA MASIVA DE INFORMACIÓN	24
3.1.3 REQUERIMIENTO LECTURA DE ACTIVOS	25
3.1.4 REQUERIMIENTO BÚSQUEDA DE ACTIVOS	25
3.1.5 REQUERIMIENTO ESTADO DE ACTIVOS	25
3.1.6 REQUERIMIENTO REGISTRO DE FOTOGRAFÍAS	26
3.1.7 REQUERIMIENTO ALMACENAMIENTO DE FOTOGRAFÍAS	26
3.1.8 REQUERIMIENTO REPORTES	26
3.2 REQUERIMIENTOS NO FUNCIONALES	27
3.2.1 INFRAESTRUCTURA	27
3.2.2 RENDIMIENTO	28
3.2.3. SEGURIDAD	28
3.2.4. DISPONIBILIDAD	29
3.2.5. MANTENIBILIDAD	29
3.2.6. PORTABILIDAD	29
3.2.7. RESTRICCIONES	30
3.2.8. LISTAS DE PARTICIPANTES	30
3.3 IDENTIFICACIÓN DE ELEMENTOS COMO SOPORTE PARA LA CREACIÓN DEL SISTEMA DE INFORMACIÓN AUTOMÁTICO.	31
3.3.1 HARDWARE Y SOFTWARE	31
3.3.2 PERFILES DE RECURSO HUMANO	32
3.3.3 DOCUMENTACIÓN	32
3.3.4 CICLO DE VIDA DEL SISTEMA	33
ETAPAS DEL PROCESO	33
□ ANÁLISIS	33
□ DISEÑO	33
□ CONSTRUCCIÓN Y PRUEBAS	33
□ IMPLEMENTACIÓN	33

4. DISEÑO DEL SISTEMA	35
4.1 MODELO ENTIDAD RELACION	35
4.2 MODELO RELACIONAL	36
4.3 DICCIONARIO DE DATOS	37
4.4 DIAGRAMA DE CLASES	40
4.5 DIAGRAMAS DE CASOS DE USO	41
4.5.1 CASO DE USO INGRESO AL SISTEMA	41
4.5.2 CASO DE USO CARGA DE INFORMACIÓN	42
4.5.3 CASO DE USO LECTURA Y BÚSQUEDA DE ACTIVO	43
4.5.4 CASO DE USO ESTADO DE ACTIVOS	44
4.5.5 CASO DE USO REGISTRO DE FOTOGRAFÍA	45
4.5.6 CASO DE USO ALMACENAMIENTO DE FOTOGRAFÍAS	46
4.5.7 CASO DE USO REPORTES	47
4.6 DIAGRAMAS DE ACTIVIDAD	48
4.6.1 DIAGRAMA DE ACTIVIDAD INGRESO AL SISTEMA	48
4.6.2 DIAGRAMA DE ACTIVIDAD CARGA DE INFORMACIÓN	49
4.6.3 DIAGRAMA DE ACTIVIDAD LECTURA Y BUSQUEDA DE ACTIVOS	50
4.6.4 DIAGRAMA DE ACTIVIDAD ESTADO DE ACTIVO	51
4.6.5 DIAGRAMA DE ACTIVIDAD REGISTRO FOTOGRÁFICO	52
4.6.6 DIAGRAMA DE ACTIVIDAD ALMACENAMIENTO DE FOTOGRAFÍA	53
4.6.7 DIAGRAMA DE ACTIVIDAD REPORTES	54
4.7 DIAGRAMAS DE SECUENCIA	55
4.7.1 DIAGRAMA DE SECUENCIA INGRESO AL SISTEMA	55
4.7.2 DIAGRAMA DE SECUENCIA CARGA DE BASE DE DATOS Y BÚSQUEDA DE INFORMACIÓN	55
4.7.3 DIAGRAMA CASO DE USO TOMAR Y ALMACENAR FOTOGRAFÍAS	56
4.7.4 DIAGRAMA DE SECUENCIA GENERAR REPORTE	56
4.7.5 HERRAMIENTAS DE DESARROLLO	57
4.7.6 DESARROLLO EN CAPAS	57
4.9 CONTROL DE FUENTES	61
5. GLOSARIO	62
6. CONSLUSIONES	
BIBLIOGRAFÍA	65

INTRODUCCIÓN

Debido al actual desarrollo tecnológico, se ve la necesidad de migrar los procesos industriales a sistemas automatizados que permitan la administración y optimización de la información en el interior de la empresa.

De acuerdo a tal necesidad, se propone una solución informática que permita a la compañía A.T.I.S. administrar la información de los inventarios con los que cuenta actualmente y optimizar la forma manual en la que se controlan los procesos a diario.

Para dar solución al presente problema se pretende hacer uso de las últimas tecnologías de información, como lo es el uso de almacenamiento en la nube y metodologías ágiles de desarrollo de software.

Así mismo se opta por soluciones en la nube debido a que las organizaciones y en general el mercado empieza a trabajar en la nube gracias a su flexibilidad y facilidad de acceso, además de los bajos costos que va a permitir a las organizaciones hoy en día automatizar sus procesos.

OBJETIVOS

OBJETIVO GENERAL

Implementar un sistema de información, factible, eficaz y robusto a la compañía A.T.I.S. que permita realizar el manejo integral de los activos fijos.

OBJETIVOS ESPECÍFICOS

- Garantizar el software bajo el concepto “Software As a Services”.
- Integrar el sistema de información en la nube, con las ventajas que ésta ofrece.
- Maximizar los márgenes de beneficio de la organización, a través de un sistema de información seguro y eficiente.
- Evaluar la operatividad técnica y tecnológica del sistema de información.

1. ASPECTOS DE LA INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

La empresa A.T.I.S. actualmente cuenta con una aplicación para la administración de inventarios, la cual se encuentra bajo una plataforma de escritorio, con bases de datos en Microsoft Access 2003 cuya velocidad y performance son muy bajos debido al alto número de transacciones y de usuarios que se emplean a diario, es por tal motivo que se hace necesario implementar una solución informática.

Gracias al constante crecimiento de la empresa A.T.I.S. y a su fortalecimiento en el mercado, hoy día se hace necesario administrar y controlar de una forma automatizada, todos los activos con los que allí se dispone, igualmente se genera la necesidad de evitar la gestión manual, la cual genera pérdida de información, inestabilidad e inseguridad en los reportes y en los procesos, como también pérdidas financieras y demás.

1.2 FORMULACIÓN DEL PROBLEMA

¿Con la creación de un Sistema de Información Integral, se dará solución al bajo rendimiento de los procesos administrativos que se realizan actualmente en el inventario de activos en la empresa A.T.I.S?

1.3 CONDICIONES DEL SISTEMA ACTUAL

El sistema de información para la administración de activos con el que cuenta la empresa Asesores En TI & Software (A.T.I.S.) en la actualidad, es un sistema semiautomático que se opera a nivel monousuario, el cual trabaja en un entorno de escritorio, bajo una plataforma Microsoft Access y un entorno de ejecución desarrollado en Visual Basic 2005, el sistema actual no cuenta con mantenimiento ni actualizaciones previas, así como tampoco dispone de una eficiencia integral.

Debido al gran número de transacciones, la aplicación presenta déficit en los tiempos de respuesta en la ejecución de sentencias y en el desarrollo de procedimientos en tiempo real, así como en el nivel de seguridad y robustez del sistema en general. Por tal razón se presenta la necesidad de implementar una solución informática que absuelva los problemas identificados en la aplicación y que permita renovar el sistema de información con el que se cuenta actualmente.

1.4 DEFINICIÓN DEL SISTEMA PROPUESTO

Implementar una aplicación web para la administración del inventario de activos en la empresa Asesores en TI & Software (A.T.I.S.), el cual operará bajo un entorno multiusuario, se desarrollará bajo la plataforma .Net (framework 4.5) y se codificará en el lenguaje C#, el motor definido para la base de datos será SQL Server 2014 Express; se han definido estas herramientas para el desarrollo de la aplicación ya que cumplen con las condiciones de rendimiento óptimas y necesarias.

La aplicación deberá generar en tiempo real las consultas y procedimientos necesarios en el levantamiento de los activos fijos, así como también debe generar la seguridad e integridad del sistema para su óptimo funcionamiento. Su arquitectura está basada en un entorno web multitarea, sobre el cual se ejecutan simultáneas transacciones y procedimientos.

El sistema a desarrollar se operará de forma simultánea por diferentes usuarios, los cuales tendrán acceso directo a la información deseada y a los procesos necesarios para el desarrollo de sus tareas, los recursos hardware para la ejecución de la aplicación están definidos dentro de un parámetro básico.

La aplicación estará hosteada en la nube debido a que permitirá a los usuarios acceder a ella de manera remota y a la compañía reducir costos de administración y licenciamiento, la aplicación se basará en un modelo SAAS (Software As A Service), la plataforma seleccionada para su publicación es Windows Azure gracias a sus beneficios en cuanto a costos para emprendimiento de software.

1.5 JUSTIFICACIÓN.

Debido a la problemática identificada en el control de inventarios dentro de la compañía, se hace necesario implementar una solución informática que permita la administración y el seguimiento de los activos en la empresa, evitando así la pérdida de información, de recursos y de tiempo en los procesos realizados.

Para implementar esta solución se ha establecido usar metodologías ágiles de desarrollo, en este caso se hará uso de la metodología Scrum, la implementación se realizará bajo plataformas Microsoft, con Visual Studio 2013 y lenguaje de desarrollo c#, Bases de datos en SQL Server y la integración se realizará bajo la plataforma de Windows Azure (nube), allí se administrará la infraestructura de la aplicación.

1.6 DELIMITACIÓN

De acuerdo al desarrollo e implantación del proyecto, se establecen las siguientes delimitaciones.

Espacial: El análisis, diseño y desarrollo del software se realizará en equipos portátiles, propiedad de los Ingenieros desarrolladores, su implementación y puesta en producción se efectuará en la nube a través de Windows Azure, desde allí se administrará la infraestructura requerida para la aplicación.

Temporal: Dentro de la planeación del proyecto, de acuerdo al cronograma se establece que el desarrollo y producción de la aplicación tendrá una duración de seis (6) meses, tiempo en el que se desarrollarán los Sprints que indica la metodología.

Conceptual: El producto software a desarrollar es un sistema de información que permite la administración de inventarios a partir de una aplicación web hosteada en la nube, la cual se conectará a una base de datos que permitirá consultar, ingresar, actualizar, y eliminar información de los activos.

1.7 ALCANCE.

El sistema consultará directamente en la base de datos cargada en la aplicación, la información de los activos se consolidará a medida que se realiza una búsqueda por serial de cada activo en la aplicación y ésta muestra los detalles de cada artículo (ubicación, estado, marca, etc.), para el desarrollo de la búsqueda se recibirá como parámetro el serial o el número de placa que identifica cada uno de los activos que hacen parte del inventario, con esta información el sistema deberá generar el reporte del inventario sobre los artículos conciliados, faltantes y sobrantes, así como también debe permitir al administrador de la aplicación realizar un seguimiento en tiempo real de las tareas desarrolladas por los demás operarios de la aplicación, igualmente debe generar un informe fotográfico de los artículos presentes en el levantamiento de la información del inventario.

El alcance del proyecto se establece de acuerdo a las necesidades del usuario, según el análisis realizado, se establece que la solución es escalable y adaptable, lo cual indica que es posible implementarla a nivel de pequeñas, medianas y grandes empresas; Las tareas y las fechas de entrega se describen en cada sprint, dentro de éstas se encuentran los siguientes entregables en base a las diferentes etapas de desarrollo del software:

- **Validación de Usuario.**
 - Autenticación del usuario mediante login y password.
 - Verificación de permisos por rol y usuario
- **Búsqueda de Activo.**
 - Búsqueda de un activo del inventario por RPMA, RPMN, o por serial.
- **Crear, editar y eliminar Activo.**
 - El aplicativo permitirá agregar un nuevo ítem al inventario.
 - Será posible editar información del activo.
 - Permitirá dar de baja a un activo con vida útil nula.
- **Detalle de Activo.**
 - El software permitirá realizar el detalle fotográfico del activo.
 - Opción de guardar y cancelar.
 - Opción de editar
- **Reportes**
 - El software deberá emitir reportes de activos, según su localización, fecha de entrega, y empleado responsable.

1.8 ESTUDIOS DE FACTIBILIDAD TÉCNICA, OPERATIVA Y ECONÓMICA

1.8.1 TÉCNICA.

La factibilidad técnica evalúa los componentes de hardware y de software asociados al desarrollo de la aplicación y de las bases de datos. Para la implementación del sistema se determina que el almacenamiento y la administración de la información se realizará bajo un ambiente web; la aplicación estará hosteada en la nube, donde ésta podrá ser accedida por los usuarios desde equipos que cuenten con el hardware necesario y básico para navegar en la internet.

La plataforma seleccionada para la implementación del sistema es Windows Azure, utilizando el servicio de máquinas virtuales en un modelo de IAAS (Infrastructure As A Service), lo cual permite la administración del software, las características de la máquina seleccionada para soportar la aplicación son 4 núcleos de CPU, 7GB de memoria RAM, 285GB de almacenamiento.

En cuanto a software esta máquina contará con sistema operativo Windows Server 2012 R2 y con servicio de servidor de aplicación IIS 8.0.

Proceso de Bases de datos

Para la administración de la base de datos se ha seleccionado SQL Azure (Microsoft SQL Server en la nube), este servicio trabaja bajo un esquema SAAS (Software As A Service) indicando que la base de datos es un servicio y como usuario, se tendrá el rol de consumidor, lo cual hace que las tareas de administración de los servidores sean hechos por parte de la plataforma y no del usuario, SQL Azure no indica las características de hardware de los servidores pero garantiza un SLA (Acuerdo de servicio, disponibilidad) del 99.5%, además garantiza que las bases de datos van a ser replicadas dentro de su infraestructura global garantizando la redundancia de información en caso de falla de uno de sus servidores.

1.8.2 OPERATIVA.

El rendimiento de la aplicación está asociado a los recursos de hardware de las máquinas seleccionadas las cuales se han especificado en la sección de factibilidad técnica, además de los planes de mantenimiento a cada una de las máquinas para garantizar un funcionamiento óptimo de la aplicación.

La disponibilidad se garantiza a partir de los SLA que ofrece Windows Azure la cual se establece en un 99.8%. Así mismos la aplicación está diseñada para que varios usuarios puedan trabajar de manera simultánea en esta, garantizando la respuesta óptima de sus operaciones.

1.8.3 ECONÓMICA.

Los costos de las aplicaciones se dividirán en, hardware, software y recursos humanos.

- ✓ Hardware: El costo de la máquina seleccionada en Azure es de 268 dólares al mes.
- ✓ Software: El costo del software necesario para correr la aplicación ya viene incluido con el costo de hardware lo cual permite un ahorro en licenciamiento.
- ✓ Recurso Humano: Se requiere un desarrollador tiempo completo con un salario de \$1.200.000 mensuales, el proyecto tiene una duración de 3 meses, por lo tanto el recurso humano tiene un costo de \$3.600.000
- ✓ Base de datos: Se va a comprar una base de datos de 10 GB inicialmente la cual tiene un costo de 10 dólares y el GB adicional 4 dólares extra.

	DESCRIPCION	TIEMPO	VALOR
HARDWARE	Máquina Azure	Mensual	\$ 536.000
SOFTWARE	No Requiere	-----	\$ 0
RECURSO HUMANO	Desarrollador de software	3 Meses	\$1.200.000
BASE DE DATOS	Almacenamiento	Mensual	\$20.000
		TOTAL	\$1.756.000

El valor total por los tres (3) meses de desarrollo del software será de \$5.268.000

2. MARCO REFERENCIAL

2.1 MARCO HISTÓRICO

Sin duda la aparición de los computadores marcó un hito dentro de los inventos registrados en la historia, de ahí en adelante comenzó una cadena sin precedentes y con un mundo infinito de posibilidades. Así, en la medida que se desarrollaba la capacidad de procesamiento de estos ordenadores y los gigantescos tamaños se reducían considerablemente, las empresas comenzaron a adquirir más computadores, el precio igualmente los hizo más accesibles y surgió definitivamente la necesidad de compartir información.

En la actualidad el manejo de la información de manera eficiente se constituye como uno de los factores de suma importancia para implementar y desarrollar estrategias competitivas. Hoy la información es administrada por los sistemas de información y las tecnologías de la información más conocidas como las TIC, las cuales han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos, suministran una plataforma de información necesaria para la toma de decisiones y, lo más importante, su implantación logra ventajas competitivas. Las tecnologías de la información han sido conceptualizadas como la integración y convergencia de la computación, las telecomunicaciones y la técnica para procesamiento de datos, donde sus principales componentes son: el factor humano, los contenidos de la información, el equipamiento, la infraestructura, el software y los mecanismos de intercambio de información, los elementos de política y regulaciones, además de los recursos financieros.

La información se ha convertido en el activo mas importante de las organizaciones, es el recurso que poseen las empresas para la toma de decisiones, las empresas han comenzado a comprender que la información no es solo un subproducto de la conducción empresarial, sino que a la vez alimenta a los negocios y puede ser uno de los factores críticos para la determinación del éxito o fracaso de las organizaciones. Para tener un mejor conocimiento sobre un sistema de información se es necesario conceptualizar esta frase, lo que se traduce en un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o un negocio. Un sistema de información realiza cuatro actividades básicas de gran importancia para una empresa, dentro de las cuales están, la entrada de información, el almacenamiento de datos, el procesamiento de información y la salida de información.

2.2 MARCO TEÓRICO

2.2.1. SCRUM

Scrum es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80. Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre 15 y 30, el equipo crea un incremento de software potencialmente entregable. El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning, durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos auto-organizados impulsando la ubicación de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto. Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

2.2.2. FASES DE SCRUM

Scrum comprende las siguientes fases:

Planificación: Definición de una nueva versión basada en la pila actual, junto con una estimación de coste y agenda. Si se trata de un nuevo sistema, esta fase abarca tanto la visión como el análisis.

Arquitectura: Diseño de la implementación de las funcionalidades de la pila. Esta fase incluye la modificación de la arquitectura y diseño generales.

Desarrollo de sprints: Desarrollo de la funcionalidad de la nueva versión con respeto continuo a las variables de tiempo, requisitos, costo y competencia. La interacción con estas variables define el final de esta fase. El sistema va evolucionando a través de múltiples iteraciones de desarrollo o sprints.

Preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la versión.

Pasos de la planificación:

- ✓ Desarrollo de un backlog completo.
- ✓ Determinación de la fecha de entrega y la funcionalidad de una o más versiones.
- ✓ Selección de la versión más adecuada para desarrollo inmediato.
- ✓ Trazado de los “paquetes del producto” (objetos) sobre los elementos del backlog de la versión elegida.
- ✓ Selección del equipo o equipos para desarrollar la nueva versión.
- ✓ Evaluación y control adecuado de los riesgos.
- ✓ Estimación del coste de la versión, incluyendo desarrollo, material, marketing, formación y despliegue.
- ✓ Conformidad de la dirección y financiación del proyecto.

Pasos de diseño y arquitectura:

- ✓ Revisión de los elementos del backlog incluidos en la versión.
- ✓ Identificación de los cambios necesarios para implementar el backlog.
- ✓ Análisis del dominio para incluir los requisitos que incluye el desarrollo mejora o actualización.
- ✓ Acotar la arquitectura del sistema para apoyar el nuevo contexto y necesidades.
- ✓ Identificar problemas del desarrollo o modificaciones.

Pasos del desarrollo (Sprint):

La fase de desarrollo es un ciclo de trabajo repetitivo. La gestión determina el cumplimiento de los tiempos, funcionalidad y calidad. Este enfoque es conocido también como ingeniería concurrente.

El desarrollo consiste en los siguientes macro-procesos:

- ✓ Reunión con los equipos para revisar los planes de lanzamiento de versión.
- ✓ Distribución, revisión y ajuste de los estándares de conformidad para el producto.
- ✓ Sprints iterativos hasta que el producto se considera listo para su distribución.

Un sprint es un conjunto de actividades de desarrollo llevado a cabo durante un periodo predefinido, por lo general entre una y cuatro semanas. Duración basada en la complejidad del producto, evaluación de riesgos y grado de supervisión deseado. El tiempo determinado para el sprint establece su velocidad e intensidad. El riesgo se evalúa de forma continua a través de las respuestas a los controles adecuados establecidos.

Cada sprint consiste en uno o varios equipos realizando:

- ✓ Desarrollo: Definición de los cambios necesarios para la implementación de los requisitos del backlog en módulos, la apertura de los módulos, análisis del dominio, diseño, desarrollo, implementación, pruebas y documentación de los cambios.
- ✓ Envoltura: Cierre de los módulos, creación de una versión ejecutable con los cambios que implementas los requisitos del backlog.
- ✓ Revisión: Reunión de todos los equipos para presentar el trabajo y revisar el progreso, identificando y resolviendo posibles cuestiones y añadiendo nuevos elementos al backlog. Se revisan los riesgos y las respuestas apropiadas.
- ✓ Ajuste: Consolidación de la información de la revisión de los módulos afectados.

Cierre

Cuando el equipo de gestión siente que las variables de tiempo, parte completada, requisitos, coste y calidad están alineadas para producir una nueva versión, declaran cerrada la versión. En esta fase se prepara el producto generado para producir una nueva versión. Entre las tareas de cierre se encuentran: integración, pruebas del sistema, documentación de usuario, preparación del material de formación y marketing.

2.3 MARCO CONCEPTUAL

2.3.1 SISTEMA DE INFORMACIÓN

Se define como un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el sistema de Información. Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

2.3.2 COMPUTACIÓN EN LA NUBE

La computación en nube es un sistema informático basado en Internet y centros de datos remotos para gestionar servicios de información y aplicaciones. La computación en nube permite que los consumidores y las empresas gestionen archivos y utilicen aplicaciones sin necesidad de instalarlas en cualquier computadora con acceso a Internet. Esta tecnología ofrece un uso mucho más eficiente de recursos, como almacenamiento, memoria, procesamiento y ancho de banda, al proveer solamente los recursos necesarios en cada momento.

2.3.3 ARQUITECTURA DE SOFTWARE

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

2.3.4 APLICACIÓN WEB

En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

2.3.5 WINDOWS AZURE

Es una plataforma ofrecida como servicio y alojada en los Data Centers de Microsoft. Windows Azure es una plataforma general que tiene diferentes servicios para aplicaciones, desde servicios que alojan aplicaciones en alguno de los centros de procesamiento de datos de Microsoft para que se ejecute sobre su infraestructura (Cloud Computing) hasta servicios de comunicación segura y federación entre aplicaciones.

2.3.6 SERVICIO WEB

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

2.3.7 MANEJADOR DE BASES DE DATOS

El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- ✓ Crear y organizar la base de datos.
- ✓ Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accesados rápidamente.
- ✓ Manejar los datos de acuerdo a las peticiones de los usuarios.
- ✓ Registrar el uso de las bases de datos.
- ✓ Interacción con el manejador de archivos: Esto a través de las sentencias en DML al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.

3. DISEÑO METODOLÓGICO

3.1 REQUERIMIENTOS FUNCIONALES

Dentro del estudio y el análisis para el desarrollo del sistema de información se identificaron los siguientes requerimientos funcionales:

3.1.1 REQUERIMIENTO DE ATENTICACIÓN Y SEGURIDAD

Número de requisito	R01		
Nombre de requisito	Autenticación y seguridad		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Ingreso al sistema y seguridad		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

El sistema deberá contar con un módulo de autenticación, mediante una combinación de usuario y contraseña los usuarios podrán acceder al sistema, así mismo estos están asociados a un rol específico el cual les permitirá tener niveles de acceso sobre la aplicación.

3.1.2 REQUERIMIENTO CARGA MASIVA DE INFORMACIÓN

Número de requisito	R02		
Nombre de requisito	Cargue masivo de información de inventario (Activos)		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Levantamiento de activos fijos		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

El sistema deberá permitir al usuario autenticado el cargue masivo de la información relacionada con el inventario (activos fijos), mediante un archivo que contenga la información relacionada con cada uno de los activos a administrar.

3.1.3 REQUERIMIENTO LECTURA DE ACTIVOS

Número de requisito	R03
Nombre de requisito	Lectura de activos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Ingreso de información a la aplicación
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Opcional Baja/

El sistema deberá permitir ingresar manualmente el serial del activo o realizar la lectura de código de barras para realizar la búsqueda del activo dentro de la información que se haya cargado previamente.

3.1.4 REQUERIMIENTO BÚSQUEDA DE ACTIVOS

Número de requisito	R04
Nombre de requisito	Búsqueda de Activos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Desarrollo del Inventario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Opcional Baja/

Posterior a la lectura del activo, el sistema deberá realizar la búsqueda de este, en caso de encontrarlo, deberá mostrar la información detallada, en caso contrario presentar un mensaje de alerta al usuario.

3.1.5 REQUERIMIENTO ESTADO DE ACTIVOS

Número de requisito	R05
Nombre de requisito	Estado de Activos (conciliados, faltantes, sobrantes)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Condición de los activos
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Opcional Baja/

El sistema deberá mostrar el resultado del activo luego de validar el número de serie de este que se obtuvo en la lectura, si se encuentra en la aplicación, dará como estado CONCILIADO, en caso de no encontrarlo, el sistema debe alertar que se reporta como FALTANTE y en caso de reportarlo físicamente y faltante en la aplicación su estado será SOBRANTE.

3.1.6 REQUERIMIENTO REGISTO DE FOTOGRAFÍAS

Número de requisito	R06		
Nombre de requisito	Registro de fotografías		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Evaluación y seguimiento de los activos		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

El sistema deberá permitir el cargue de fotografías para relacionarlas a los activos fijos de la empresa en el proceso de evaluación y seguimiento.

3.1.7 REQUERIMIENTO ALMACENAMIENTO DE FOTOGRAFÍAS

Número de requisito	R07		
Nombre de requisito	Almacenamiento de fotografías		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Toma de fotografías		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

El sistema deberá almacenar las fotografías en archivos consecutivos de los activos.

3.1.8 REQUERIMIENTO REPORTES

Número de requisito	R08		
Nombre de requisito	Reportes		
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción		
Fuente del requisito	Evaluación y seguimiento de los activos		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

El sistema deberá generar reportes del estado de los activos de la empresa. Reporte de activos e historial fotográfico.

3.2 REQUERIMIENTOS NO FUNCIONALES

3.2.1 INFRAESTRUCTURA

Debido a la naturaleza de la computación en la nube, Windows Azure ofrece una gama de combinaciones de hardware con sistemas operativos ya reconstruidos para la aplicación web siguiendo un modelo IAAS (Infrastructure As A Service).

Aplicación web

Se ha seleccionado la máquina virtual de tamaño medio con las características de hardware que se ven en la siguiente tabla.

Virtual machine size	CPU cores	Memory	OS disk space—cloud services	OS disk space—virtual machines	Max. data disks (1 TB each)	Max. IOPS (500 per disk)
ExtraSmall	Shared	768 MB	19 GB	20 GB	1	1x500
Small	1	1.75 GB	224 GB	70 GB	2	2x500
Medium	2	3.5 GB	489 GB	135 GB	4	4x500
Large	4	7 GB	999 GB	285 GB	8	8x500
ExtraLarge	8	14 GB	2,039 GB	605 GB	16	16x500
A6	4	28 GB	999 GB	285 GB	8	8x500
A7	8	56 GB	2,039 GB	605 GB	16	16x500

Esta máquina viene preinstalada con sistema operativo Windows Server 2012. Como software adicional se necesita que tenga habilitado como servidor de aplicaciones el Internet Information Services 8.0.

Base de datos

Azure ofrece la base de datos SQL Server como servicio en la nube (SAAS) lo cual hace que no sea necesario conocer las características de hardware y software de estas, para ello la plataforma simplemente entrega una dirección de acceso y ofrece dos tipos de base de datos dependiendo de su tamaño, para el sistema actual se optó por una base de datos de 15GB inicialmente.

SQL Database is available in the following editions:

- **Web and Business:** Web and Business offer relational databases run on shared resources, with built-in replicas within a datacenter. Web edition supports up to 5GB databases, and Business edition supports up to 150 GB databases. Both editions support dynamic scale out of thousands of distributed databases.
- **Premium** (preview): Built on the same foundation as Web and Business, Premium offers access to built-in business-class functionality that is easy to dial up and dial down based on need. The marquee capability of Premium is reserved capacity. A reservation guarantees a fixed amount of capacity that is not shared with any other database, and thus helps deliver more powerful performance. Premium is currently in preview, try it today.

3.2.2 RENDIMIENTO

- El sistema deberá soportar un entorno multiusuario.
- Para evitar demoras en el procesamiento de la información, el sistema deberá aceptar fotografías máximo de 3MB.
- Se deberá tener un motor de base de datos y un servidor de aplicaciones de contingencia en caso de fallas en los principales.

3.2.3. SEGURIDAD

- El motor de base de datos seleccionada solo podrá ser accedida desde la aplicación web y desde la maquina donde se desarrolle el sistema.
- El sistema deberá contar con un módulo de log, este se encargara de almacenar el historial de todas las transacciones que hagan los usuarios sobre la aplicación.
- No deberá permitir el inicio de sesión del mismo usuario más de una vez de manera simultánea.
- La información relacionada con autenticación de usuarios (usuarios, contraseñas) deberán estar cifradas en la base de datos.

3.2.4. DISPONIBILIDAD

SLA (Acuerdo de Servicio): El sistema deberá estar disponible un 99.4% del tiempo mes, y en caso de caída no debe durar más de 4 horas fuera de servicio.

3.2.5. MANTENIBILIDAD

En caso de mantenimiento que requiera dar de baja el sistema por un tiempo determinado, se deberá avisar con anticipación a los usuarios correspondientes con al menos una semana de anticipación y las horas fuera de línea no deben superar las permitidas según el SLA.

Se realizará un mantenimiento mensual a la aplicación para mejorar performance, reiniciando el servidor de aplicaciones IIS y los pool relacionados con este.

Para la base de datos, el mantenimiento consistirá en la reconstrucción de índices y optimización de tablas de sistema, adicionalmente se diseñará un sistema de sincronización de datos para mantener una copia local de este.

3.2.6. PORTABILIDAD

En caso de que desee migrar el sistema a otra plataforma, el sistema nuevo deberá contener las mismas o características superiores de hardware, adicionalmente el servidor de aplicaciones debe contar con el .NET Framework 4.0 o superior y base de datos SQL Server 2012.

3.2.7. RESTRICCIONES

Tipo de restricción	Descripción de la Restricción
Avalúo del inventario	Para este desarrollo NO se realizará el proceso de avalúo económico sobre el inventario financiero de la empresa, el desarrollo de esta fase se realizara como se ejecuta actualmente.
Software	El sistema en su primera versión de desarrollo será únicamente compatible con plataformas Microsoft en ambiente web.

3.2.8. LISTAS DE PARTICIPANTES

Nombre	Lina María Cubillos Vergara
Rol	Analista y Desarrolladora de software
Categoría profesional	Ingeniera de Sistemas
Responsabilidades	Análisis del sistema de Información y diseño del mismo
Información de contacto	Lina_mariia@hotmail.com

Nombre	Jorge Eliecer Vargas
Rol	Director de Proyecto
Categoría profesional	Ingeniero de Sistemas
Responsabilidades	Revisión y dirección del proyecto

3.3 IDENTIFICACIÓN DE ELEMENTOS COMO SOPORTE PARA LA CREACIÓN DEL SISTEMA DE INFORMACIÓN AUTOMÁTICO.

3.3.1 HARDWARE Y SOFTWARE

Especificaciones de las máquinas de desarrollo para el sistema.

HARDWARE	<ul style="list-style-type: none">✓ Memoria RAM: 4GB o superior✓ Procesador: Intel Core i3 2GHZ✓ Disco duro: 250GB o superior
SOFTWARE	<ul style="list-style-type: none">✓ Sistema Operativo Microsoft Windows (7, 8) 32 bits o 64 bits✓ Entorno de desarrollo Visual Studio 2012.✓ Lenguaje de Desarrollo C#✓ Motor Base de Datos SQL Server 2012 Express Edition,

3.3.2 PERFILES DE RECURSO HUMANO

Para el desarrollo del aplicativo web se hace necesario contar con profesionales del área informática quienes establezcan las métricas para el desarrollo del proyecto, ingenieros de sistemas con capacidad de análisis para el diseño y construcción del sistema de acuerdo a los requerimientos exigidos por parte del usuario final, de igual forma contribuye al desarrollo del proyecto un ingeniero industrial, quien documente y evalúe la calidad del sistema propuesto.

Perfil ingeniero de Desarrollo con conocimientos en:

- ✓ Desarrollo bajo plataforma .Net lenguaje C#
- ✓ Desarrollo de aplicaciones web con ASP.NET
- ✓ Manejo de base de datos con SQL Server

3.3.3 DOCUMENTACIÓN

Al final del proyecto se generara la siguiente documentación:

- ✓ Documento de requerimientos: Se detallaran cada uno de los requerimientos con los flujos de estos.
- ✓ Documento de casos de uso: Gráficamente se representaran todos los flujos de cada uno de los requerimientos con los actores asociados a este.
- ✓ Diccionario de Datos: especificación de las entidades, campos y objetos de la base de datos a implementar en el sistema.
- ✓ Manual de Usuario: este manual está dirigido al usuario final del sistema, básicamente es el uso de cada uno de los módulos de la aplicación.

3.3.4 CICLO DE VIDA DEL SISTEMA

Para el proceso de desarrollo se elige una metodología ágil ya que permitirá entregas rápidas al cliente y la verificación corta de cada uno de los componentes que se van a desarrollar, se ha elegido scrum ya que su proceso iterativo (sprints) se adaptan a los requerimientos definidos anteriormente, adicionalmente scrum se caracteriza por ser incremental lo cual permitirá que cada uno de los componentes que se vayan añadiendo al sistema.

Etapas del proceso

✓ **Análisis**

Se realizan el afianzamiento de las especificaciones funcionales del sistema, requerimientos de hardware, software, restricciones, planes de mantenimiento.

Entregable: Documento de alcance y casos de uso.

✓ **Diseño**

Construcción del modelo entidad relación (MER), diseño de interfaz de usuario e integración de componentes. Diseño de los sprints para cada uno de los entregables del sistema.

Entregable: Modelo entidad relación, diccionario de datos, diagramas de casos de uso, diagramas de secuencia.

✓ **Construcción y Pruebas**

Desarrollo de todos los componentes del software según los sprints definidos, integración de componentes de software, construcción de objetos de bases de datos, documentación técnica, pruebas de usabilidad y carga de información.

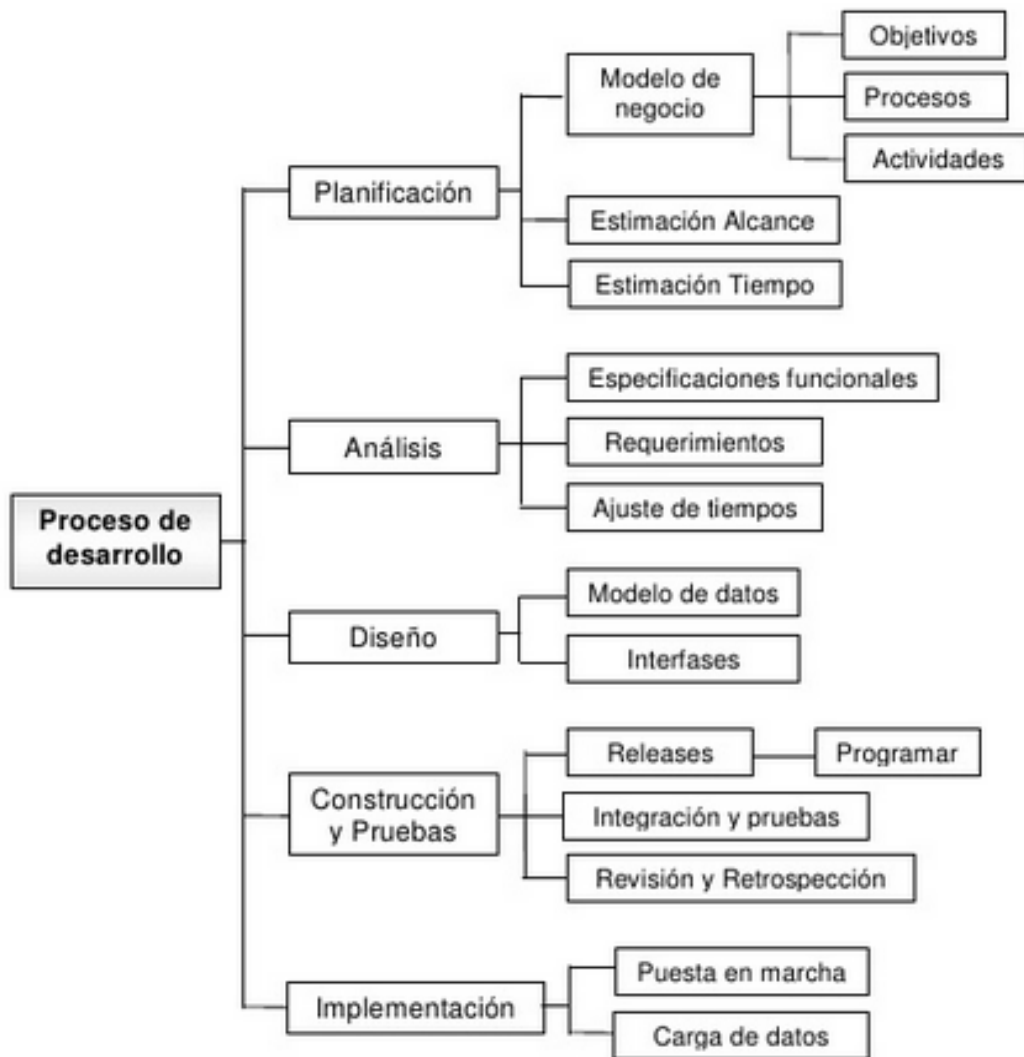
Entregable: Sistema funcional con resultados de pruebas.

✓ **Implementación**

Puesta en marcha de la aplicación y la base de datos en los ambientes dispuestos para su puesta en producción, construcción de manuales de usuario, pruebas de conectividad y acceso externo por parte de los usuarios, capacitación a usuario final.

Entregable: Sistema implementado en el ambiente de producción, manual de usuario.

Diagrama de división de trabajo

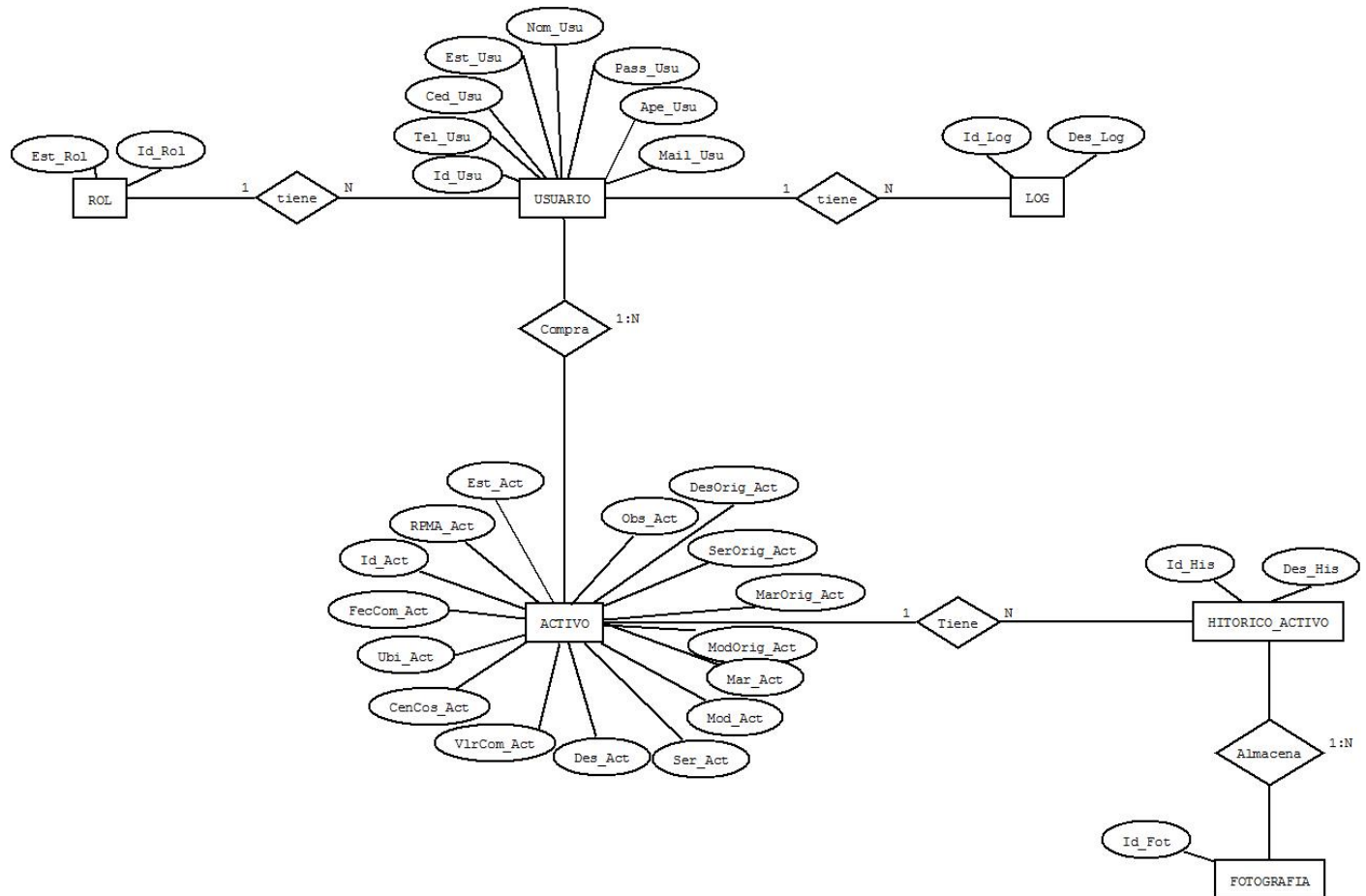


1

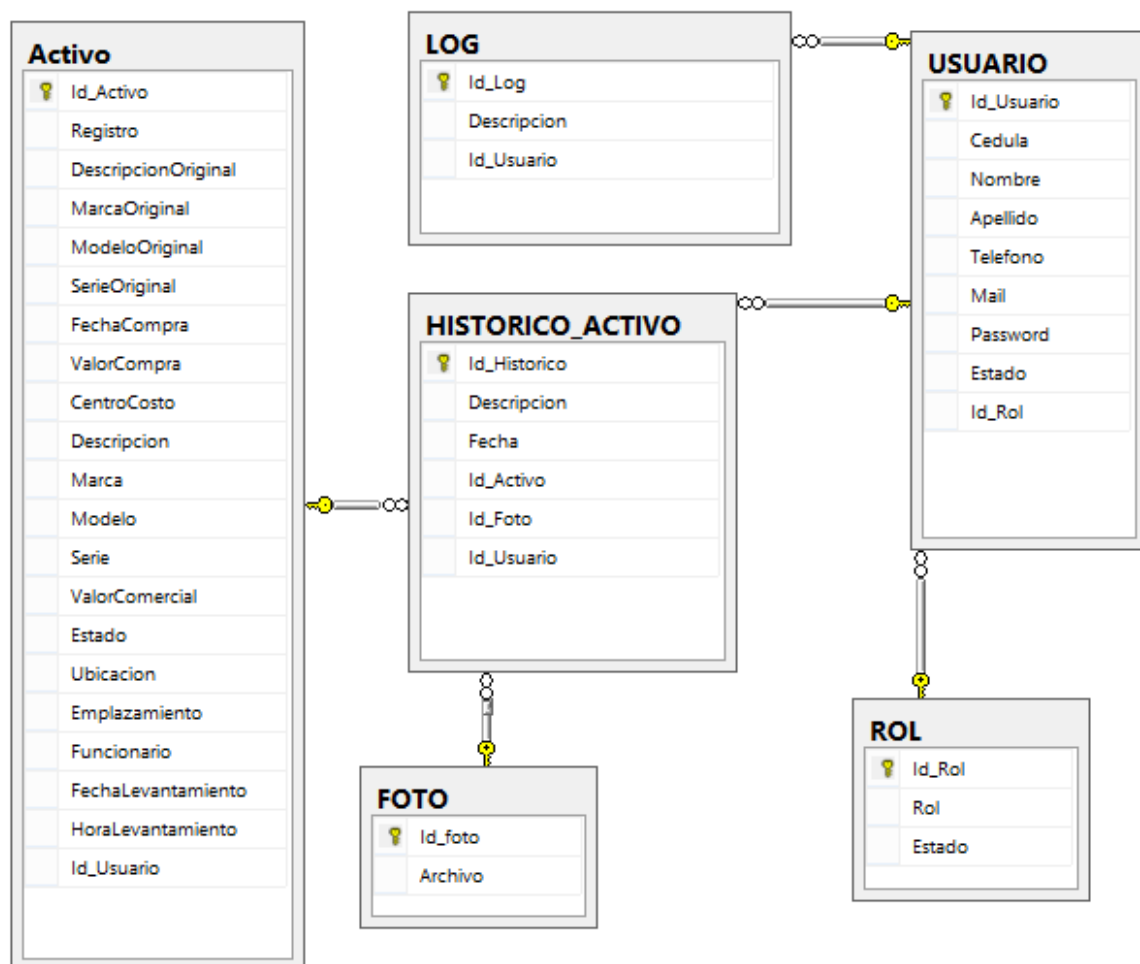
¹“Método Ágil scrum aplicado al desarrollo de software”, María Laura Citon, Universidad de Mendoza.
Página 43

4. DISEÑO DEL SISTEMA

4.1 MODELO ENTIDAD RELACION



4.2 MODELO RELACIONAL



4.3 DICCIONARIO DE DATOS

Las entidades que conforman el sistema, se detallan y se describen a continuación:

❖ Entidad Usuario

Columna	Tipo de Dato	Tamaño	Es Nulo	PK	FK	Descripción
Id_Usu	Nchar	10	No	Si	No	Identificación del usuario
Ced_Usu	Nchar	10	Si	No	No	Cedula del usuario
Nom_Usu	Varchar	50	Si	No	No	Nombre del usuario
Ape_Usu	Varchar	50	Si	No	No	Apellido del usuario
Tel_Usu	Int		Si	No	No	Teléfono del usuario
Mail_Usu	Varchar	50	Si	No	No	e-mail del usuario
Pass_Usu	Varchar	50	Si	No	No	Contraseña del usuario
Est_Usu	Varchar	50	Si	No	No	Estado del usuario
Id_Rol	Nchar	10	Si	No	Si	Llave foránea de la tabla Rol

❖ Entidad Log

Columna	Tipo de Dato	Tamaño	Es Nulo	PK	FK	Descripción
Id_Log	Nchar	10	No	Si	No	Identificación del log
Des_Log	varchar	50	Si	No	No	Descripción del log
Id_Usu	Nchar	10	Si	No	Si	Llave foránea de la tabla Usuario

❖ Entidad Fotografía

Columna	Tipo de Dato	Tamaño	Es Nulo	PK	FK	Descripción
Id_Fot	Nchar	10	No	Si	No	Identificación de la fotografía
Id_Act	Nchar	10	Si	No	Si	Llave foránea de la tabla Activo

❖ Entidad Historico_Activo

Columna	Tipo de Dato	Tamaño	Es Nulo	PK	FK	Descripción
Id_His	Nchar	10	No	Si	No	Identificación del histórico
Des_His	Nchar	10	Si	No	No	Descripción del Histórico
Id_Act	Nchar	10	Si	No	Si	Llave foránea de la tabla Activo.
Id_Fot	Nchar	10	Si	No	Si	Llave foránea de la tabla Fotografía

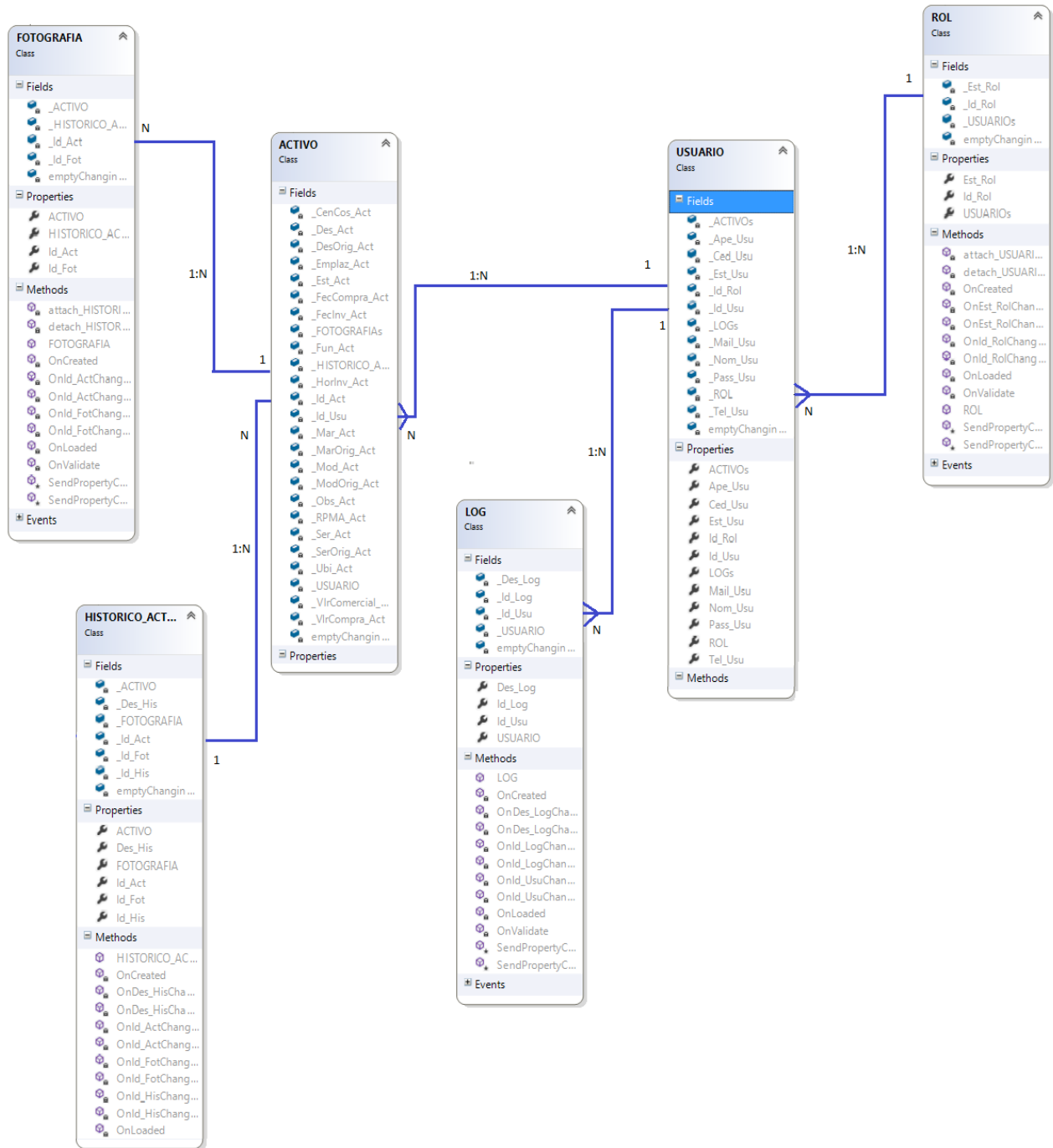
❖ Entidad Rol

Columna	Tipo de Dato	Tamaño	Es Nulo	PK	FK	Descripción
Id_Rol	Nchar	10	No	Si	No	Identificación del rol
Est_Rol	varchar	50	Si	No	No	Estado del Rol

❖ Entidad Activo

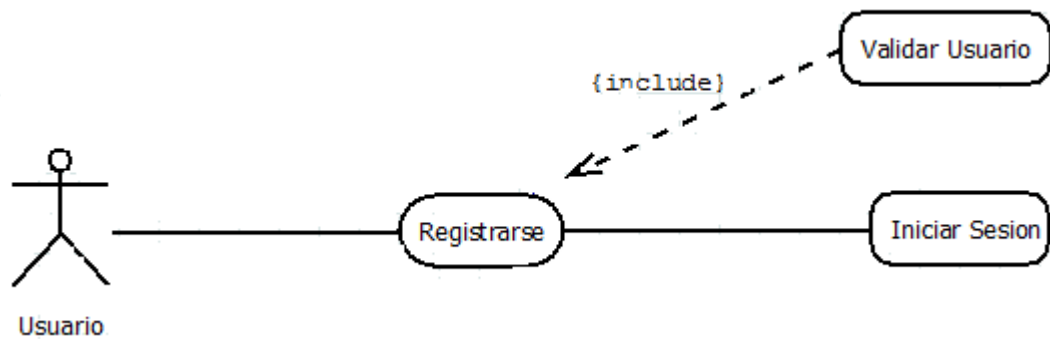
Columna	Tipo de Dato	Tamaño	Es Nul o	PK	FK	Descripción
Id_Act	Int		No	Si	No	Identificación del activo
RPMA_Act	varchar	50	Si	No	No	Registro para monitoreo Antiguo del activo
DesOrig_Act	varchar	50	Si	No	No	Descripción original del activo
MarOrig_Act	varchar	50	Si	No	No	Marca original del activo
ModOrig_Act	varchar	50	Si	No	No	Modelo original del activo
SerOrig_Act	Date		Si	No	No	Serie original del activo
FecCompra_Act	Int		Si	No	No	Fecha de compra del activo
VlrCompra_Act	varchar	50	Si	No	No	Valor de compra del activo
CenCos_Act	varchar	50	Si	No	No	Centro de costo del activo
Des_Act	varchar	50	Si	No	No	Descripción del activo
Mar_Act	varchar	50	Si	No	No	Marca del activo
Mod_Act	varchar	50	Si	No	No	Modelo del activo
Ser_Act	varchar	50	Si	No	No	Serie del activo
Obs_Act	Int		Si	No	No	Observaciones
VlrComercial_Act	varchar	50	Si	No	No	Valor comercial del activo
Est_Act	varchar	50	Si	No	No	Estado del activo
Ubi_Act	varchar	50	Si	No	No	Ubicación del activo
Emplaz_Act	varchar	50	Si	No	No	Emplazamiento del activo
Fun_Act	varchar	50	Si	No	No	Funcionario responsable del inventario
FecInv_Act	Time		Si	No	No	Fecha del levantamiento del inventario
HorInv_Act	nchar	10	Si	No	No	Hora del levantamiento del inventario
Id_Usu	nchar	10	Si	No	Si	Identificación Usuario

4.4 DIAGRAMA DE CLASES



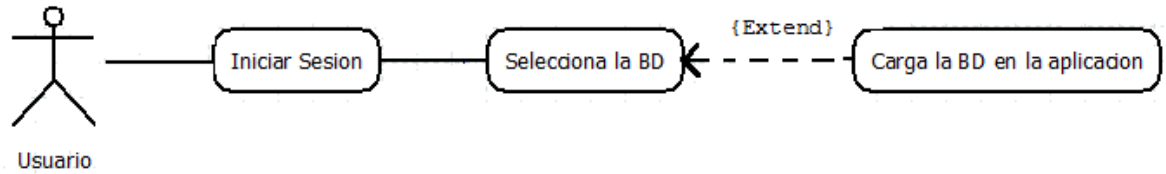
4.5 DIAGRAMAS DE CASOS DE USO

4.5.1 CASO DE USO INGRESO AL SISTEMA



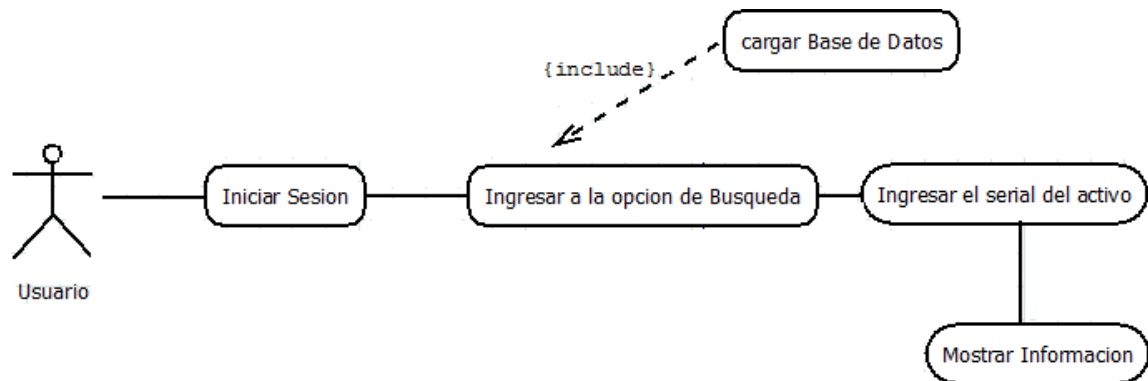
Identificador:	CU – [R01]
Nombre Caso de Uso:	Ingreso al sistema.
Descripción:	Caso de uso para la autenticación ante el sistema.
Actor(es):	Usuario no autenticado.
Flujo Normal:	<ol style="list-style-type: none">1) El usuario introduce su nombre y contraseña.2) Clic en el botón enviar.3) Ingresa al perfil de usuario.
Flujo Alternativo:	<ol style="list-style-type: none">1) El usuario introduce su nombre y contraseña.2) Clic en el botón enviar.3) En caso de que los datos sean incorrectos, la aplicación se redirige a la página de login. <ol style="list-style-type: none">1) EL usuario hace clic en el botón enviar.2) El sistema debe presentar el mensaje “Usuario y contraseña obligatorios”.
Entrada:	Datos de usuario (nombre y contraseña)
Salida:	Autenticación en el sistema y visualización del perfil si lo datos son correctos.

4.5.2 CASO DE USO CARGA DE INFORMACIÓN



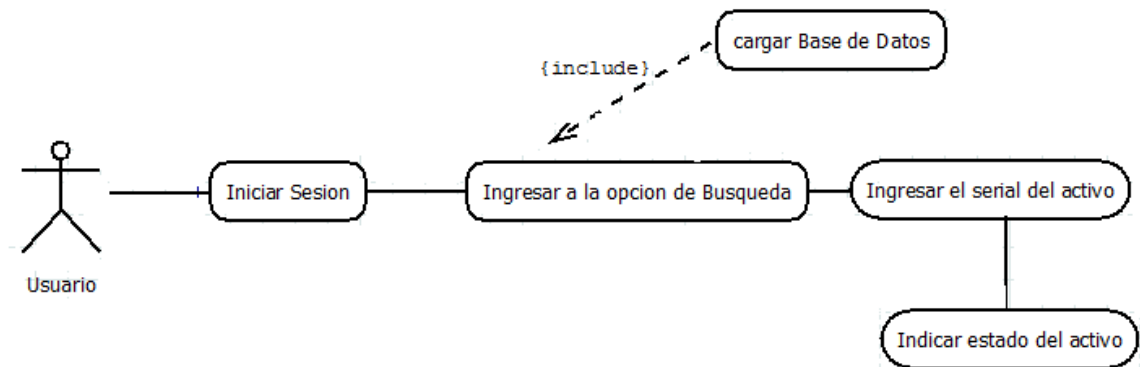
Identificador:	CU – [R02]
Nombre Caso de Uso:	Carga de información.
Descripción:	Caso de uso para ingresar la base de datos del cliente a la aplicación.
Actor(es):	Usuario autenticado.
Flujo Normal:	1) El usuario inicia sesión. 2) Clic en el módulo Bases de Datos. 3) Clic en el botón cargar base de datos.
Flujo Alternativo:	1) El usuario inicia sesión. 2) Clic en el módulo Bases de Datos. 3) Clic en el botón cargar base de datos. 4) En caso de no encontrar la base de datos la aplicación genera un mensaje de alerta y se redirige al módulo de base de datos.
Entrada:	Base de Datos del usuario
Salida:	

4.5.3 CASO DE USO LECTURA Y BÚSQUEDA DE ACTIVO



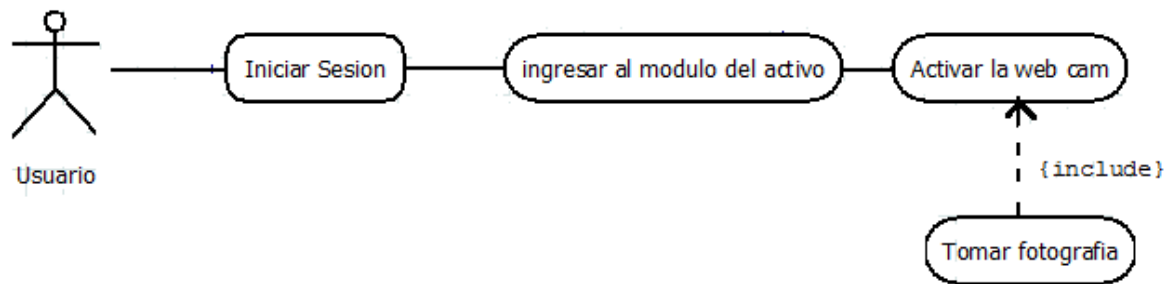
Identificador:	CU – [R03]
Nombre Caso de Uso:	Búsqueda de Activos.
Descripción:	Caso de uso para buscar activos dentro del inventario.
Actor(es):	Usuario autenticado.
Flujo Normal:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) Clic en la opción Búsqueda. 3) El usuario ingresa manualmente o a través de código de barras el serial del activo. 4) La aplicación debe mostrar toda la información referente al activo.
Flujo Alternativo:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) Clic en la opción Búsqueda. 3) El usuario ingresa manualmente o a través de código de barras el serial del activo, en caso de no encontrar el activo, la aplicación debe generar un mensaje de alerta.
Entrada:	Serial del activo
Salida:	Información detallada del activo.

4.5.4 CASO DE USO ESTADO DE ACTIVOS



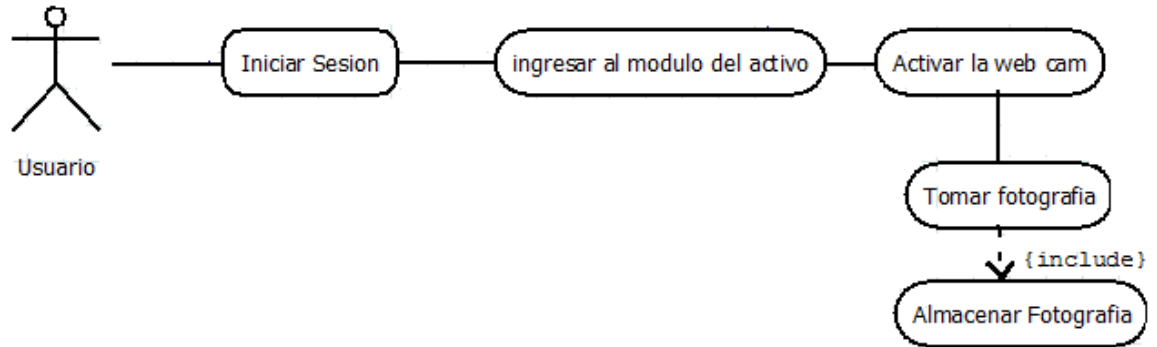
Identificador:	CU – [R04]
Nombre Caso de Uso:	Estado de los Activos.
Descripción:	Caso de uso para indicar el estado de los activos.
Actor(es):	Usuario autenticado.
Flujo Normal:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) Clic en la opción Búsqueda. 3) El usuario ingresa manualmente o a través de código de barras el serial del activo. 4) La aplicación debe mostrar el estado del activo.
Flujo Alternativo:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) Clic en la opción Búsqueda. 3) El usuario ingresa manualmente o a través de código de barras el serial del activo. 4) En caso de no encontrarse el activo, el sistema debe generar un mensaje de alerta.
Entrada:	Serial del activo
Salida:	Estado del activo.

4.5.5 CASO DE USO REGISTRO DE FOTOGRAFÍA



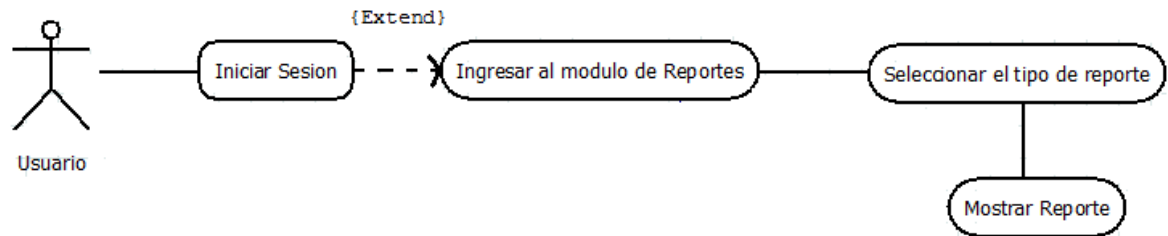
Identificador:	CU – [R05]
Nombre Caso de Uso:	Registro de Fotografía.
Descripción:	Caso de uso para indicar el estado de los activos.
Actor(es):	Usuario autenticado.
Flujo Normal:	1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Inventarios. 3) Selecciona la opción de cargar fotografía.
Flujo Alternativo:	1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Inventarios. 3) Selecciona la opción de cargar fotografía. 4) En caso de no cargar la fotografía el sistema debe arrojar alertar el error.
Entrada:	Serial del Activo
Salida:	Fotografía del activo.

4.5.6 CASO DE USO ALMACENAMIENTO DE FOTOGRAFÍAS



Identificador:	CU – [R06]
Nombre Caso de Uso:	Almacenamiento de Fotografía.
Descripción:	Caso de uso para indicar el estado de los activos.
Actor(es):	Usuario autenticado.
Flujo Normal:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Inventarios. 3) Selecciona la opción de cargar fotografía. 4) Se almacena la fotografía en una ruta específica.
Flujo Alternativo:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Inventarios. 3) Selecciona la opción de cargar fotografía. 4) Se almacena la fotografía en una ruta específica. 5) En caso de no tomar la fotografía el sistema debe arrojar alertar el error.
Entrada:	Serial del Activo
Salida:	Fotografía almacenada del activo.

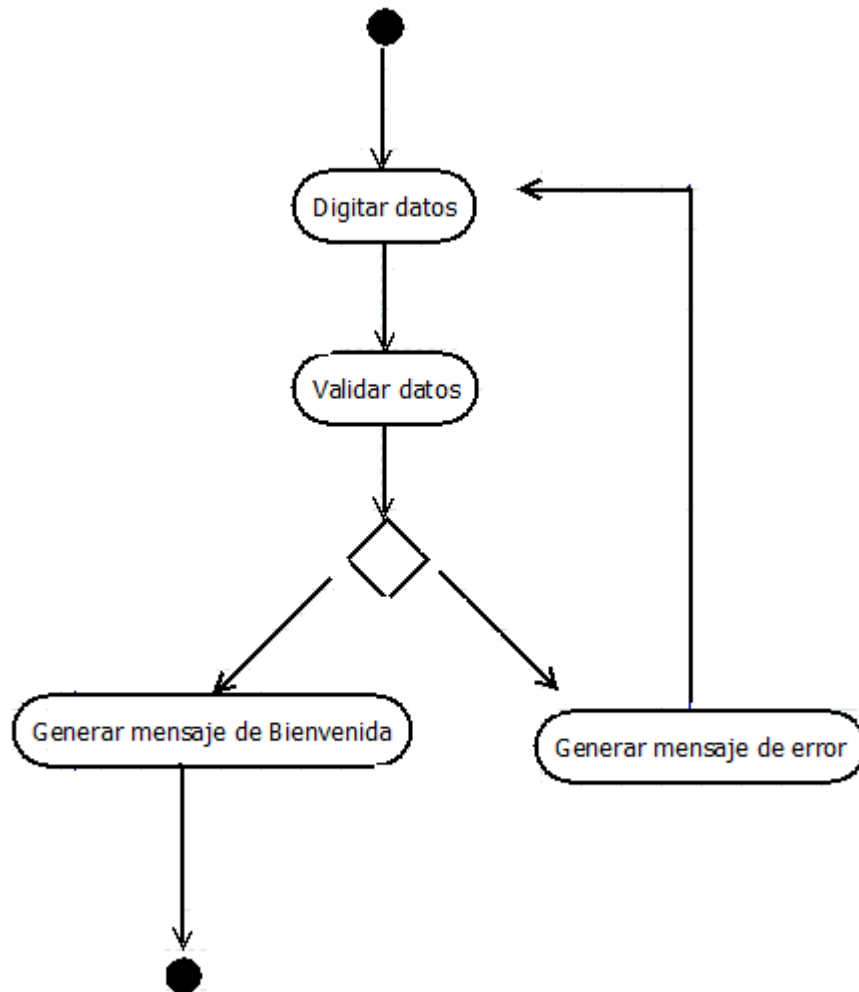
4.5.7 CASO DE USO REPORTES



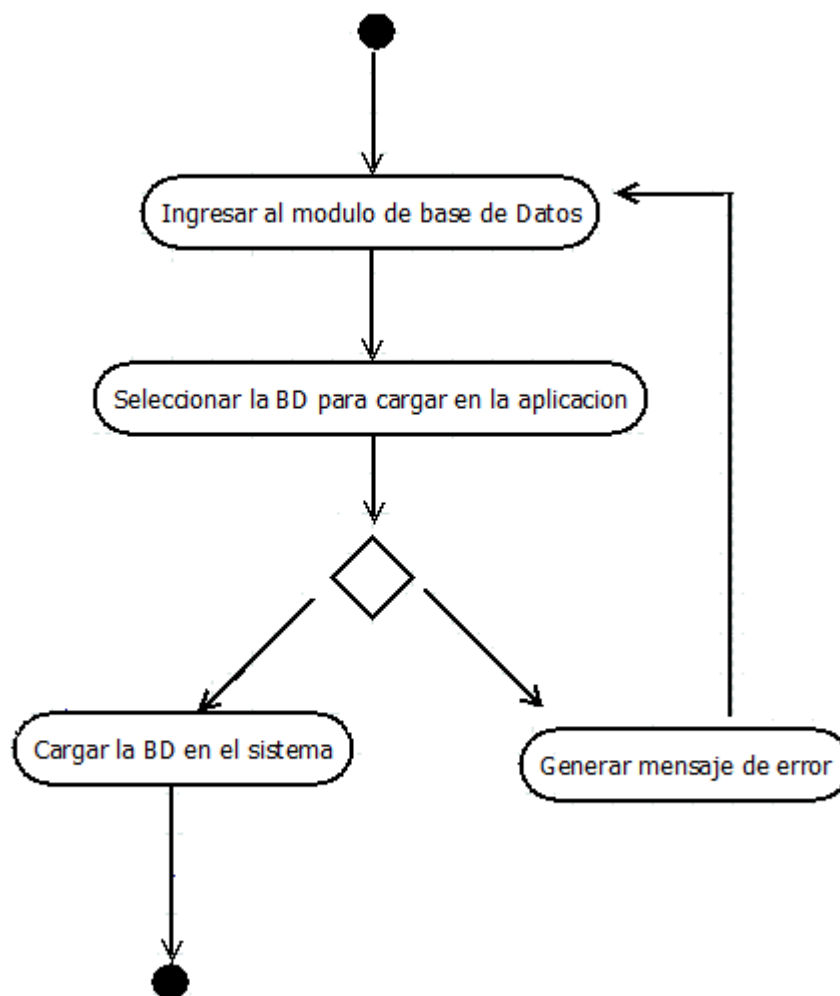
Identificador:	CU – [R07]
Nombre Caso de Uso:	Reportes.
Descripción:	Caso de uso para solicitar reportes.
Actor(es):	Usuario autenticado.
Flujo Normal:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Reportes. 3) Selecciona el tipo de reporte deseado. 4) Clic en tomar Generar Reporte.
Flujo Alternativo:	<ol style="list-style-type: none"> 1) El usuario inicia sesión. 2) El usuario ingresa al módulo de Reportes. 3) Selecciona el tipo de reporte deseado. 4) Clic en tomar Generar Reporte. 5) En caso de no generarse el reporte, el sistema debe generar un mensaje de alerta.
Entrada:	ID del usuario
Salida:	Reporte solicitado.

4.6 DIAGRAMAS DE ACTIVIDAD

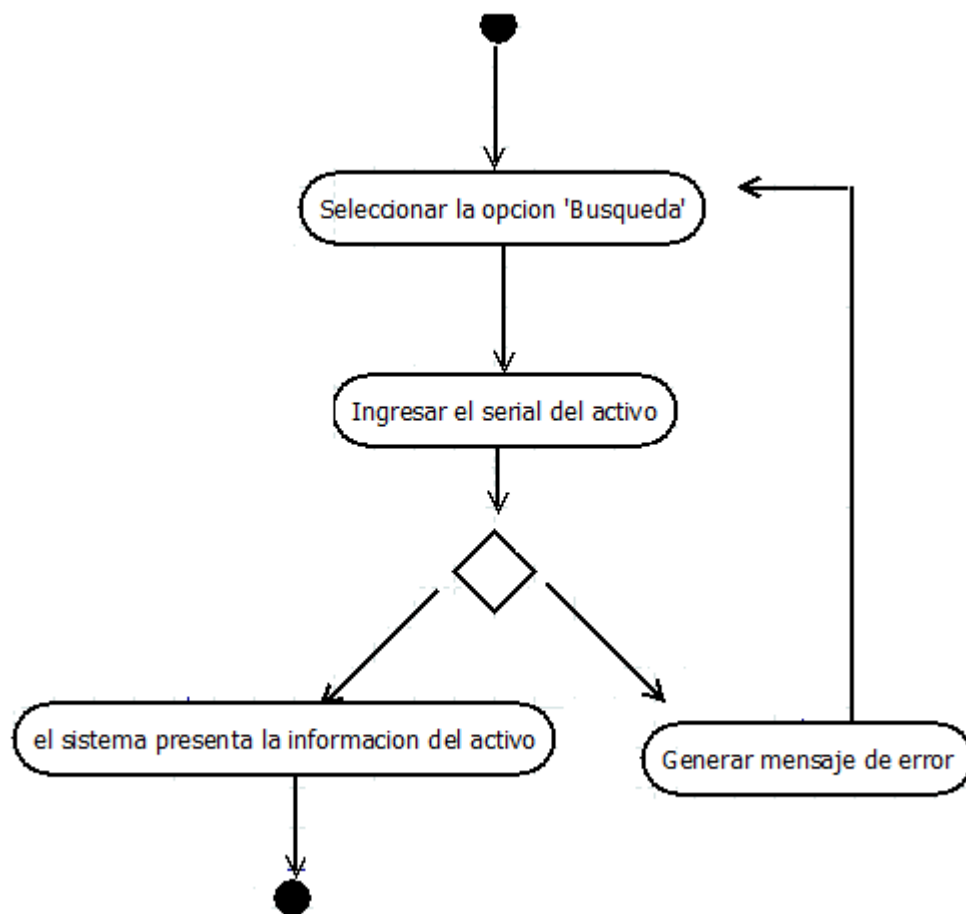
4.6.1 DIAGRAMA DE ACTIVIDAD INGRESO AL SISTEMA



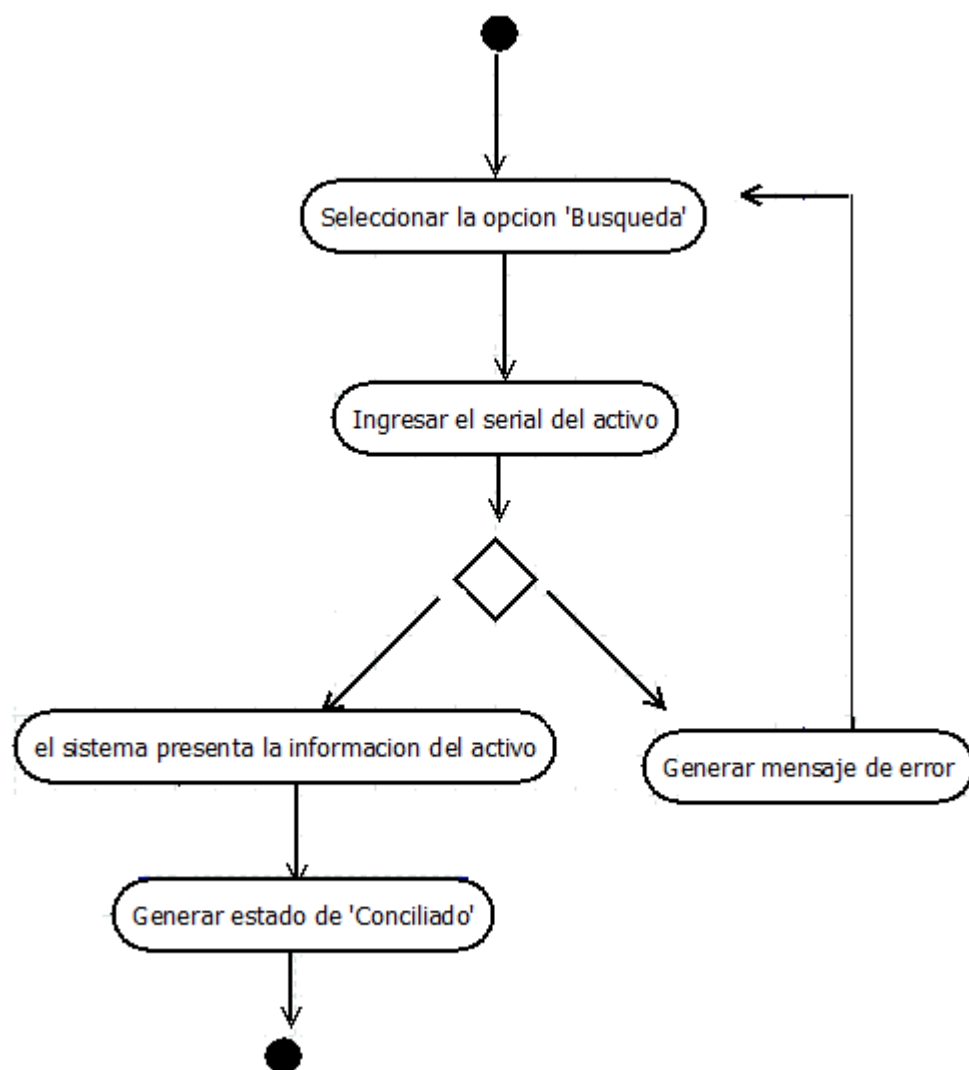
4.6.2 DIAGRAMA DE ACTIVIDAD CARGA DE INFORMACIÓN



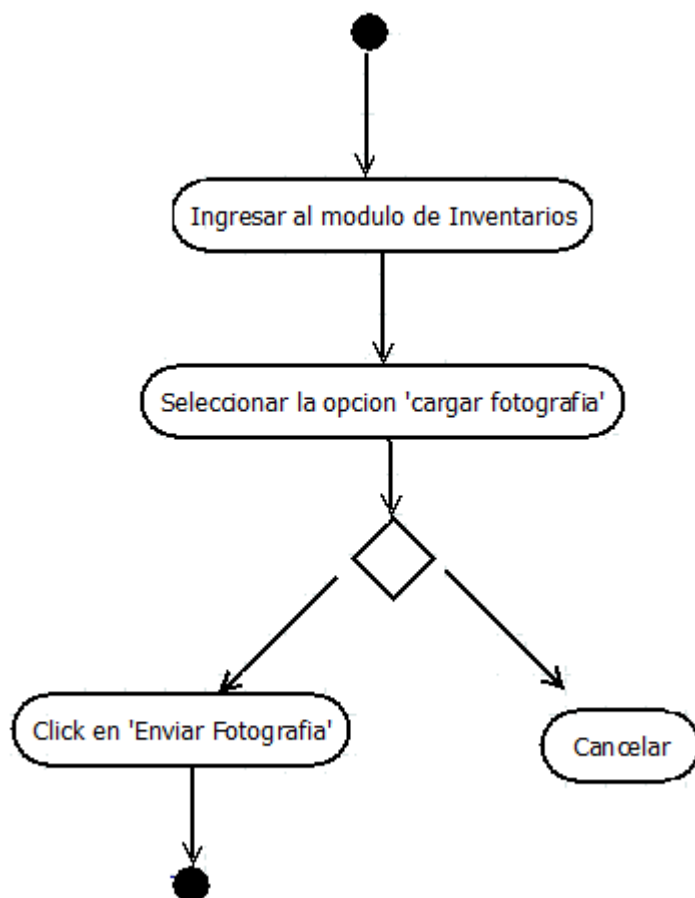
4.6.3 DIAGRAMA DE ACTIVIDAD LECTURA Y BUSQUEDA DE ACTIVOS



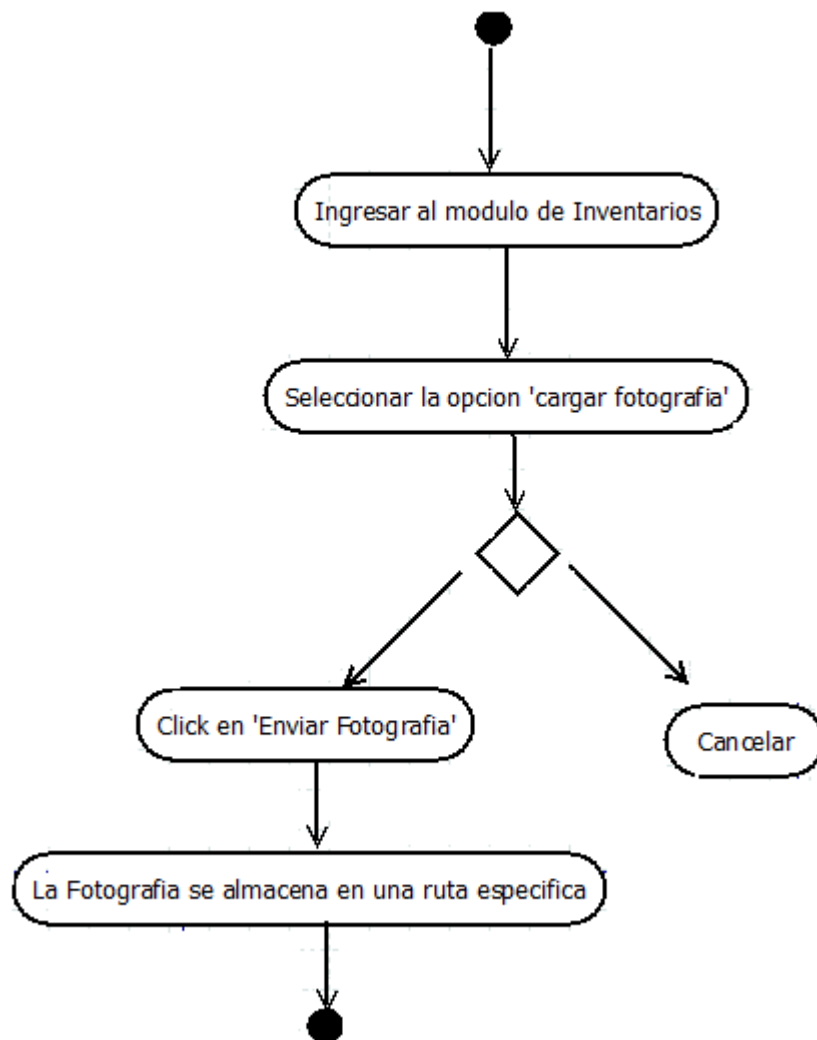
4.6.4 DIAGRAMA DE ACTIVIDAD ESTADO DE ACTIVO



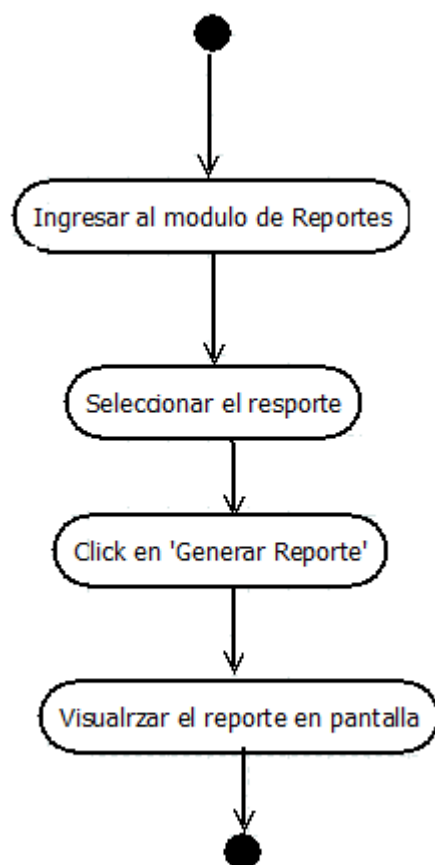
4.6.5 DIAGRAMA DE ACTIVIDAD REGISTRO FOTOGRÁFICO



4.6.6 DIAGRAMA DE ACTIVIDAD ALMACENAMIENTO DE FOTOGRAFÍA

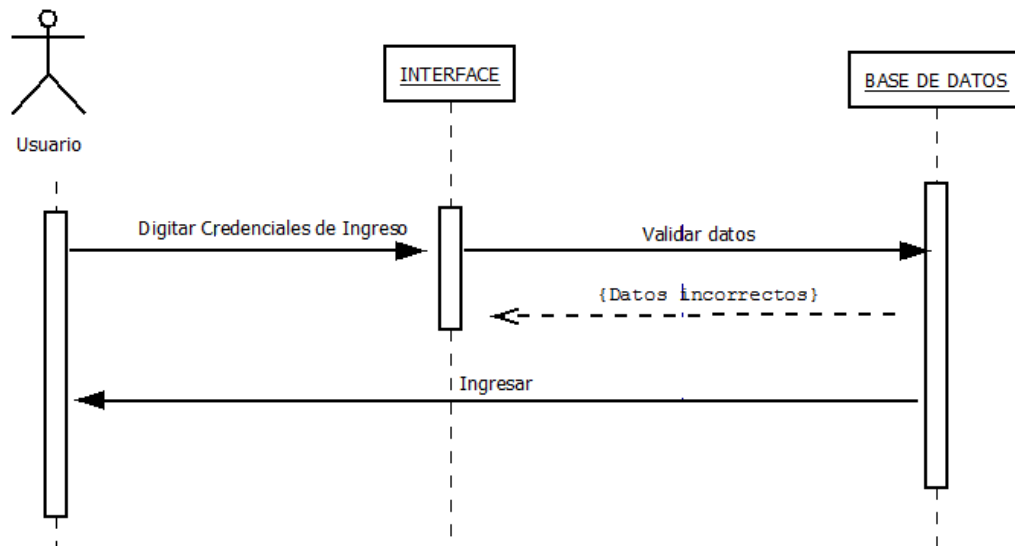


4.6.7 DIAGRAMA DE ACTIVIDAD REPORTES

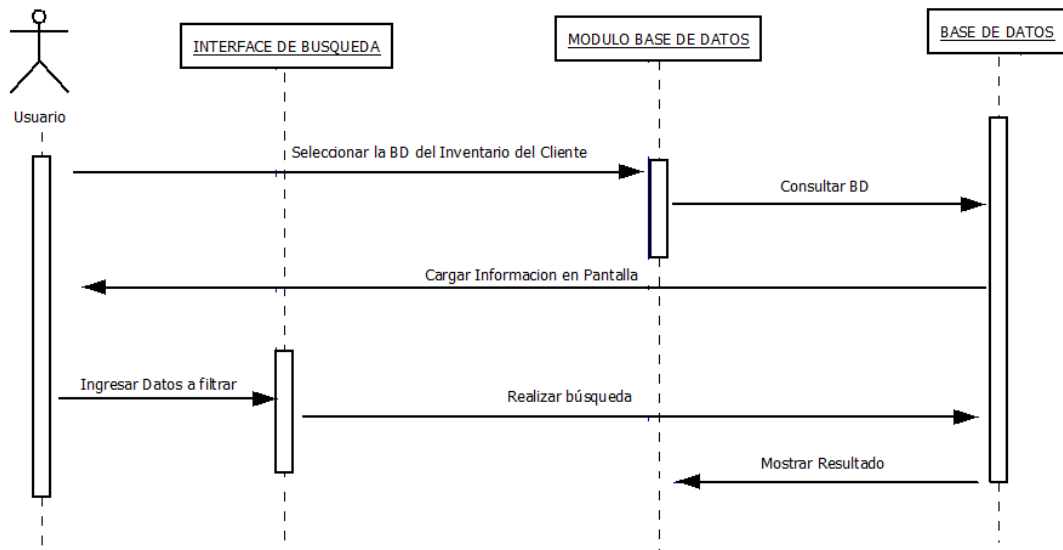


4.7 DIAGRAMAS DE SECUENCIA

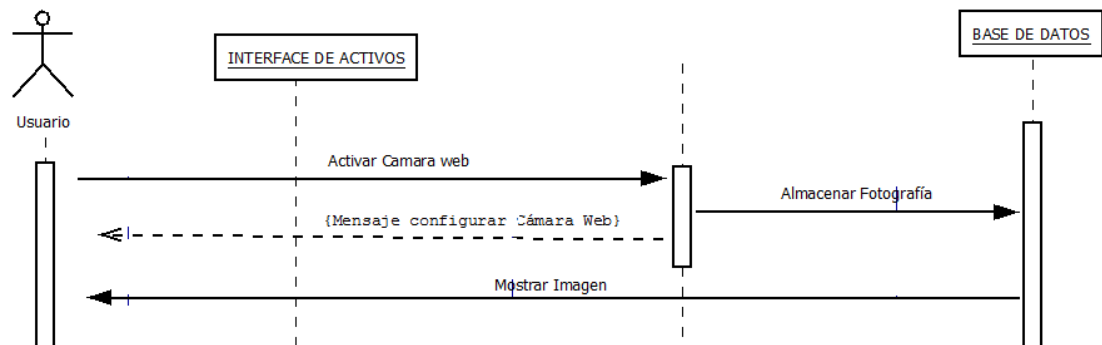
4.7.1 DIAGRAMA DE SECUENCIA INGRESO AL SISTEMA



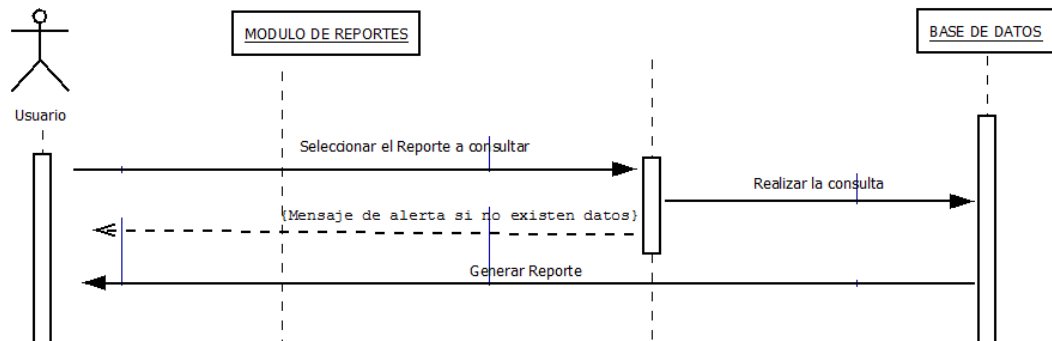
4.7.2 DIAGRAMA DE SECUENCIA CARGA DE BASE DE DATOS Y BÚSQUEDA DE INFORMACIÓN



4.7.3 DIAGRAMA CASO DE USO TOMAR Y ALMACENAR FOTOGRAFÍAS



4.7.4 DIAGRAMA DE SECUENCIA GENERAR REPORTE



ARQUITECTURA DEL SISTEMA

4.7.5 HERRAMIENTAS DE DESARROLLO

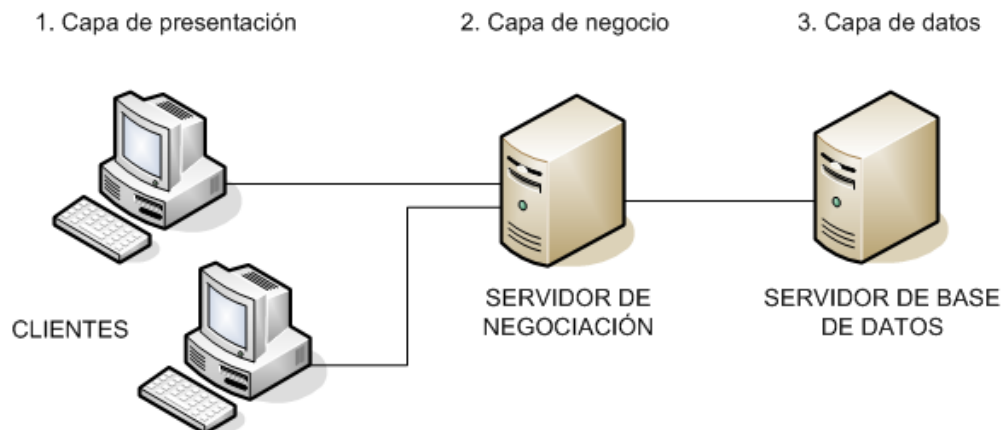
Las herramientas de desarrollo a utilizar son:

- IDE de desarrollo: Visual Studio 2013 Express Edition for Web.
- Versión del Framework: 4.5 incluida con Visual Studio.
- Motor de base de datos: SQL Server 2014 Express.
- Control de Versiones: Team Foundation Service (Visual Studio Online).

4.7.6 DESARROLLO EN CAPAS

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se aborda el nivel requerido sin tener que revisar entre código mezclado.

El Sistema se ha diseñado mediante una arquitectura basada en capas, éstas permiten una abstracción de los diferentes componentes del sistema (acceso a datos, servicios, objetos, presentación, etc.), y también facilitan las tareas de administración, mantenimiento y escalabilidad del sistema, además permite una fácil integración con componentes de terceros quienes deseen interactuar con la aplicación.



Para este sistema se ha seleccionado una arquitectura a 3 capas descritas a continuación:

4.7.6.1 CAPA DE ACCESO A DATOS (Data Access Layer)

En esta capa residen los datos. Está formada por uno o más gestor de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

La capa de acceso a datos está constituida por todos los componentes de software encargados de comunicarse con la fuente de datos, por buenas prácticas de desarrollo, esta capa no debe contener sentencias o instrucciones SQL, debe incluir el llamado de funciones y/o procedimientos almacenados enviando sus correspondientes parámetros y tipos de dato solicitados.

Para el sistema se ha implementado como componente de conexión a datos, la herramienta LINQ to SQL, la cual permitirá mapear los objetos de base de datos a código administrado (C#), esto restringirá a que el desarrollador tenga acceso a cadenas de conexión, e instrucciones SQL ya diseñadas y construidas en la base de datos.

Esta capa de acceso a datos se ha definido dentro de la solución de desarrollo con la nomenclatura Tesis.Inventario.DAL.

4.7.6.2 CAPA DE LÓGICA DE NEGOCIO (Business Logical Layer)

La capa de negocio encapsula toda la lógica de la aplicación, es aquí donde se establecen todas las reglas que deben cumplirse, Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Esta capa de negocio se ha definido dentro de la solución de desarrollo con la nomenclatura Tesis.Inventario.BLL.

*Por buena práctica de desarrollo se ha decidido ubicar gran parte de la lógica de negocio en los procedimientos almacenados y funciones de la base de datos, esto debido a las capacidades de hardware del servidor de base de datos, y asimismo con el fin de evitar múltiples llamados a esta capa desde la aplicación en un módulo específico.

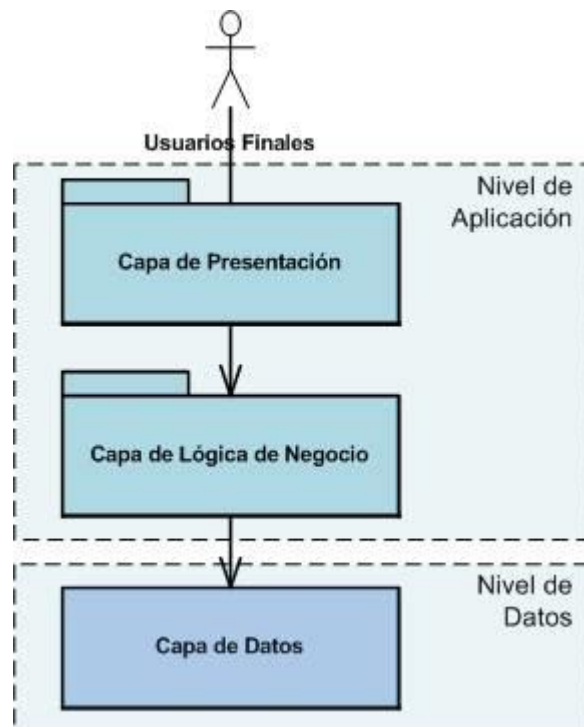
4.7.6.3 Capa de Presentación

También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario, contiene todos los elementos referentes a la aplicación web en desarrollo, paginas, imágenes, iconos, hojas de estilos y demás. Esta capa se comunica únicamente con la capa de negocio.

La capa de presentación se ha definido dentro de la solución de desarrollo con la nomenclatura Tesis.Inventario.Web.

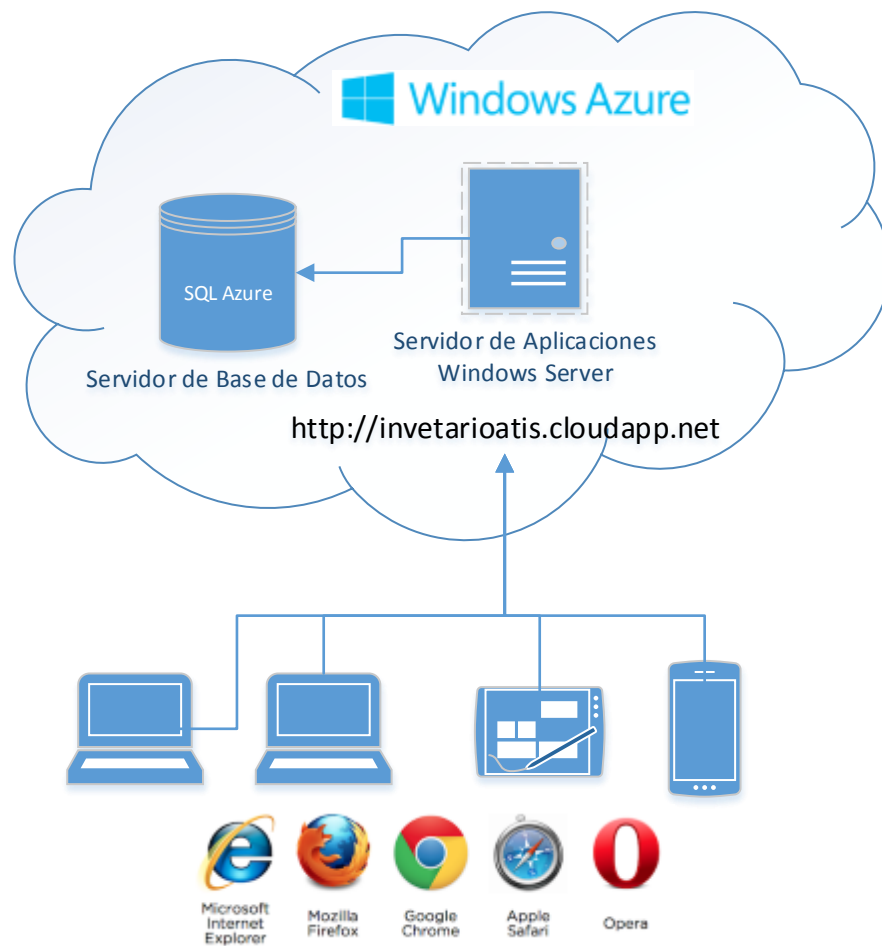
4.7.6.4 Interacción entre Capas

Uno de los principios del desarrollo por capas es la comunicación, cada una de las capas tiene un objetivo específico y por lo tal debe establecer interfaces de comunicación con otras capas que deseen acceder a ella. La arquitectura a 3 capas es la definida para el sistema. La comunicación entre las capas se verá representada de la siguiente manera.



4.8 DESPLIEGUE

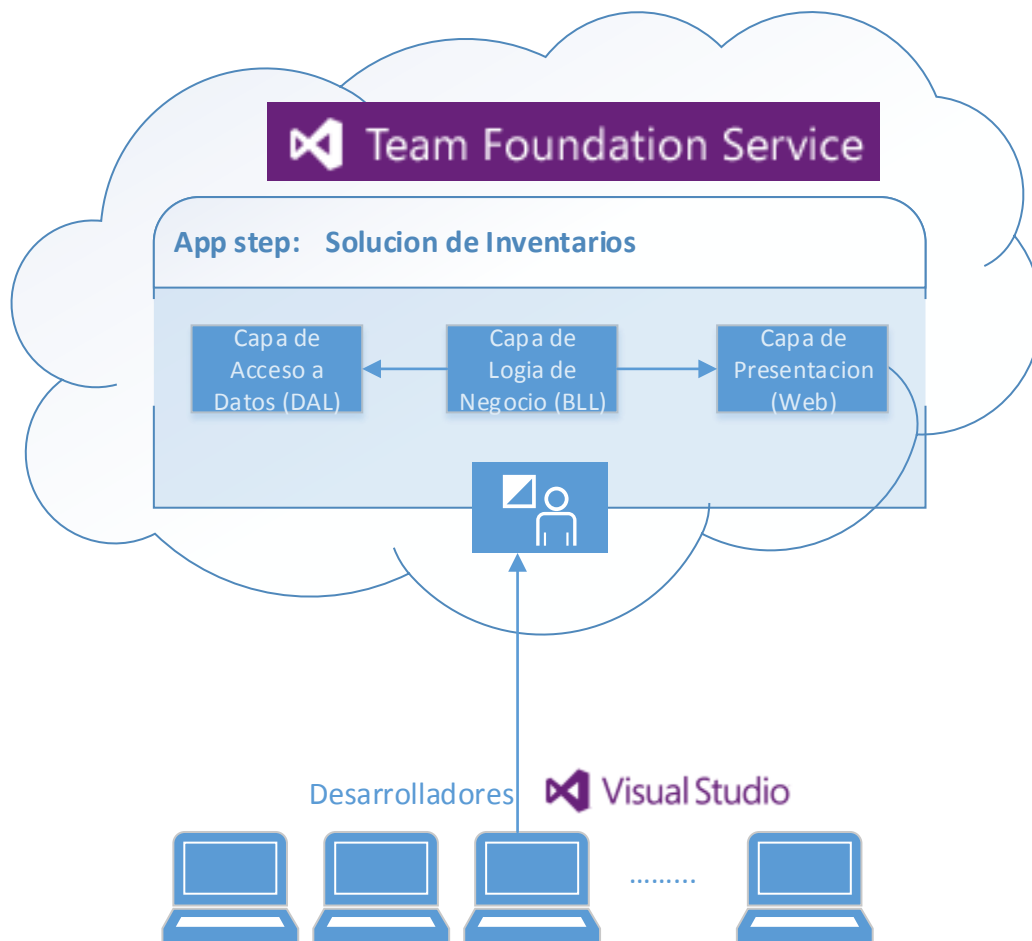
Como se definió en el numeral 3.2.1 (Infraestructura) del presente documento la aplicación va a ser llevada a la nube mediante un modelo SAAS evitando los gastos de hardware e infraestructura por parte de la compañía y del usuario final.



El despliegue de la solución suministrará los servidores necesarios tanto para la base de datos como para la aplicación web. El objetivo final será que sin importar el tipo de dispositivo o plataforma se pueda acceder al sistema de forma rápida y sencilla, para ello Windows Azure fue seleccionado como plataforma de despliegue debido a su alta confiabilidad, ágil administración y fácil escalamiento de recursos en caso de ser requerido.

4.9 CONTROL DE FUENTES

El servidor de código fuente seleccionado ha sido Team Foundation Service, ya que es una herramienta factible y altamente eficaz, ya que al estar en la nube permite no solo tener versionamiento fuera de las instalaciones o servidores de la compañía y de las máquinas del equipo de desarrollo, sino también permitirá al equipo acceder a los componentes de la aplicación desde cualquier pc con conexión a internet.



5. GLOSARIO

.NET: es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

SaaS: (“Software as a Service”) o software como servicio, es un término que se usa para describir cuando los usuarios "rentan" o usan prestado un software en línea, en lugar de comprarlo e instalarlo en sus propios equipos de hardware.

IaaS: (“Infrastructure as a Service”) es una plataforma virtual en la que se requiere de un entorno operativo y aplicaciones o servicios desplegables. Es un modelo de disposición en el que una organización externaliza la capacidad de proceso y almacenamiento. Son utilizados para operaciones, incluidos servidores, sistemas de almacenamiento, conexiones, enrutadores entre otros.

WINDOWS AZURE: Es una plataforma de nube abierta y flexible que permite compilar, implementar y administrar aplicaciones rápidamente, en una red global de centros de datos administrados por Microsoft. Puede compilar aplicaciones en cualquier lenguaje, herramienta o marco. Y puede integrar sus aplicaciones de nube públicas con el entorno de TI existente.

INTERNET INFORMATION SERVER (IIS): Es una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente.

BASE DE DATOS SQL AZURE: Base de datos SQL de Windows Azure, antes SQL Azure, es un servicio de base de datos relacional completamente administrado que proporciona capacidad de administración flexible, incluye alta disponibilidad integrada, ofrece rendimiento predecible y admite el escalado masivo.

SLA: Un acuerdo de nivel de servicio o Service Level Agreement, también conocido por las siglas ANS o SLA, es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio. El ANS es una herramienta que ayuda a ambas partes a llegar a un consenso en términos del nivel de calidad del servicio, en aspectos tales como tiempo de respuesta, disponibilidad horaria, documentación disponible, personal asignado al servicio, etc.

TEAM FOUNDATION SERVER: Es la plataforma de colaboración para equipos de Desarrollo de software, permite gestionar todo el ciclo de desarrollo desde el análisis de requerimientos hasta el despliegue final de una aplicación en servidores de producción con componentes como control de código fuente, servidor de compilaciones, reportes, portal de equipo para colaboración, seguimiento de proyecto, etc.

CONCLUSIONES

Con el uso de metodologías ágiles como SCRUM se genera mayor flexibilidad a los proyectos de software, ya que se anticipan los cambios durante el desarrollo, el cliente permanece actualizado en cuanto a los avances del proyecto, y para los desarrolladores el ambiente de trabajo es más independiente y menos ambiguo.

Con la implementación de SAAS (Software as a Services) se genera una menor inversión inicial y se minimizan los riesgos, el poder utilizar el software sin tener que realizar una inversión inicial en máquinas y software para el funcionamiento de la aplicación es un beneficio importante para los directores de IT y en general para las empresas.

En la actualidad los sistemas de información que se desarrollan bajo tecnologías en la nube no solo ayudan a mejorar procesos internos en las compañías, también contribuyen a la reducción de costos y recursos ecológicos de las empresas. Así mismo el uso de herramientas en la nube facilita el acceso a los insumos sin depender de un punto de red, ubicación geográfica o maquina física para continuar con el correcto desarrollo del proyecto.

La arquitectura de software, el análisis y el diseño en los proyectos de software, son pautas esenciales para el buen desarrollo y conclusión de aplicaciones informáticas, ya que permiten a los proyectos concluir satisfactoriamente, les otorga escalabilidad y adaptabilidad así como también un óptimo mantenimiento y robustez en el futuro.

La implementación de soluciones informáticas en la nube es una gran oportunidad de robustez, seguridad e integridad para las compañías que se integran a esta tecnología, además de ser un nuevo auge en la industria tecnológica y una nueva oportunidad de escalabilidad y portabilidad económica para las diferentes compañías.

6. BIBLIOGRAFÍA

- López Román Leobardo, Metodología de programación orientada a objetos, ed Alfa Omega 2006
- Pressman, Roger. Ingenieroa de software un Enfoque Práctico. Editorial MacGraw Hill. Cuarta Edición. 1998 México.
- Senn, James A Analisis y diseño de sistemas de información 2° Ed. Mexico, MacGraw Hill 1992

RECURSOS EN LA WEB

- Pastor Sánchez, Juan Antonio. Construcción de portales: Curso Los portales de Internet. Facultad de Ciencias de la Documentación, España: Universidad de Murcia. <http://gti1.edu.um.es:8080/portales/PORTALES-construccion.pdf>

HERRAMIENTAS UTILIZADAS

- Visual Studio 2013 Express for Web: <http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>
- Visual Studio Online: Team Foundation Service
<http://www.visualstudio.com/>
- SQL Server 2014 Express: <http://msdn.microsoft.com/en-us/library/dn434042.aspx>